MEDICAL IMAGE REGISTRATION AND SURGERY SIMULATION

IMM-PHD-1996-25

Morten Bro-Nielsen

July 26, 1997

Preface

This thesis has been prepared at the Department of Mathematical Modelling (IMM), Technical University of Denmark. It is a partial fulfillment of the requirements for the degree of Ph.D. in engineering.

The subject is Medical Image Registration and Surgery Simulation, with a specific focus on the physical models used in these fields.

Many of the techniques, described and developed in the thesis, are on the forefront of technology and, therefore, do not always have practical applications today. But, I hope the reader will read the thesis with an open mind, and enjoy the perspectives which some of the techniques open up for the future of technology in medicine.

Reading this thesis requires a basic knowledge of image analysis, computer graphics, physical models, and the nomenclature used in medical image analysis.

Lyngby, August 1996

Morten Bro-Nielsen

Acknowledgements

The work reported in this thesis has been carried out at three different locations: The Department of Mathematical Modelling (IMM) at the Technical University of Denmark, the Epidaure group at INRIA, France, and the 3D-Lab at the School of Dentistry, University of Copenhagen, Denmark.

It was during the 9 months I spent in the Epidaure group, that the main basis and ideas of this thesis were developed. I would, therefore, like to thank dr. Nicholas Ayache, very much, for giving me the chance to work there. It was a source of continuous inspiration and great fun to interact with the members of the group, and I am grateful to everybody there. In particular, I would like to thank dr. Hervé Delingette who never tired of answering my questions on Euler-Lagrange equations, and Stephane Cotin with whom I shared many fruitful discussions on surgery simulation.

After I returned to Denmark, I have spent most of my time at the 3D-Lab in Copenhagen. The 3D-Lab is a joint research center for the National University Hospital of Denmark, the School of Dentistry, University of Copenhagen, and IMM at the Technical University of Denmark.

Professor Sven Kreiborg of the School of Dentistry has been the driving force behind the 3D-Lab. I would like to thank Sven for allowing me to work there, and for the many insights into medicine and dentistry, which he has given me. In addition, I am grateful to Per Larsen and especially Tron Darvann for their tremendous help with practical issues.

During the Ph.D., I have been the supervisor for four M.Sc. thesis students. Working with these students has been a source of inspiration for much of the work, which is in this thesis. I would, therefore, like to thank Rolf Jackson, Michael Konstantinidis, Bo Rasmussen, and especially Claus Gramkow, for the effort, and energy they put into their work - and also the fun we had together.

Dr. Mads Nielsen from the department of Computer Science, the University of Copenhagen, has also been a great inspiration throughout the project. His explanations of theoretical aspects of scale-space and regularization, both orally and in papers, have been important for my basic understanding of physical models.

I would like to thank my supervisors, professor Knut Conradsen and dr. Bjarne Kjær Ersbøll, and the rest of the Image Analysis Group at IMM for the chance to work on this Ph.D. thesis. In particular, I would like to thank Knut for always backing me up, whenever I got into trouble. Also the TAP people

at IMM, especially Helle Welling, Finn Kuno Christensen, Else Johansen, and Ruth Bredsdorff, deserve credit for their loyal support and help with all the practical things.

My sister Helle Bro-Nielsen helped tremendously during the last hectic week, with proof-reading and correction of the manuscript. Many thanks to her also.

This thesis was mainly supported by the Technical University of Denmark, and partly by the Danish Research Academy and Knud Højgårds Foundation.

For typesetting the LaTeX 2_ε package has been used.

This thesis is dedicated to KMB.

Summary

This thesis explores the application of physical models in medical image registration and surgery simulation. The continuum models of elasticity and viscous fluids are described in detail, and this knowledge is used as a basis for most of the methods described here.

Rigid image registration using voxel similarity measures are reviewed, and new measures based on Grey Level Cooccurrence Matrices (GLCM) are introduced. These measures are evaluated extensively using CT, MR, and cryosection images from the Visible Human data set. The results show that mutual information remains the best generally applicable measure. But for specific modality combinations the new GLCM measures show considerable promise.

Results of registering the CT image to the red channel of the cryosection image, and the CT image to the MR image are shown.

A new and faster algorithm for non-rigid registration using viscous fluid models is presented. This algorithm replaces the core part of the original algorithm with multi-resolution convolution using a new filter, which implements the linear elasticity operator. Using the filter results in a speedup of at least an order of magnitude. Use of convolution hardware is expected to improve the performance even more.

Non-rigid registration using a physically valid model of bone growth is also presented. Using medical knowledge about the growth processes of the mandibular bone, a registration algorithm for time sequence images of the mandible is developed. Since this registration algorithm models the actual development of the mandible, it is possible to simulate the development.

Finally, real-time deformable models, using finite element models of linear elasticity, are developed for surgery simulation. The time consumption of the finite element method is reduced dramaticly, by the use of condensation techniques, explicit inversion of the stiffness matrix, and the use of selective matrix vector multiplication.

Reviews of both medical image registration and surgery simulation work are given.

Keywords: Medical image registration, rigid registration, non-rigid registration, grey level cooccurrence matrices, Visible Human data set, voxel similarity measures, physical models, continuum models, elastic models, viscous fluid models, linear elasticity, convolution, elastic registration, fluid

registration, bone growth, growth models, virtual reality, surgery simulation, cranio-facial surgery simulation, finite element models, condensation, linear matrix systems.

Resumé

Denne afhandling behandler anvendelsen af fysiske modeller i medicinsk billedregistrering og operationssimulering. Kontinuummodeller for elasticitet og viskose fluider beskrives detaljeret, og viden om disse modeller danner basis for de fleste af de metoder der beskrives i afhandlingen.

Rigid billedregistrering vha. voxel-similaritetsmål beskrives, og nye mål baseret på Grey Level Cooccurrence Matrices (GLCM) introduceres. Disse mål testes vha. CT, MT, og histologiske snitbilleder fra Visible Human databasen. Resultaterne viser at mutual information forbliver det bedste generelt anvendelige mål til rigid registrering. Men, for enkelte modalitetskombinationer viser de nye GLCM mål lovende muligheder.

Resultater vises for registrering af CT billede til den røde kanal af det histologiske snitbillede, og CT billede til MR billede.

En ny og hurtigere algoritme til ikke-rigid registrering vha. viskose fluide modeller præsenteres. Denne algoritme erstatter den centrale del af den originale algoritme, med multi-skala konvolution vha. et nyt filter, som implementerer den lineært elastiske operator. Ved at anvende dette filter opnås en hastighedsforøgelse på mindst en faktor. Med specialiseret konvolutions-hardware kan endnu hurtigere behandling forventes.

Ikke-rigid registrering vha. en fysisk korrekt model af knoglevækst beskrives også. Med medicinsk viden om vækstprocesserne i mandiblen, udvikles en registreringsalgoritme til registrering af tidssekvensbilleder af mandiblen. Siden registreringsalgoritmen modellerer den egentlige udvikling af mandiblen, er det muligt at simulere denne udvikling.

Endelig udvikles real-tids deformerbare modeller til operationssimulering vha. finite element modeller af lineær elasticitet. Disse modellers tidsforbrug reduceres kraftigt vha. kondenseringsteknikker, eksplicit invertering af stivhedsmatricen, og anvendelsen af selektiv matrix vektor multiplicering.

Reviews af både medicinsk billedregistrering og operationssimulering gives.

Nøgleord: Medicinsk billedregistrering, rigid registrering, ikke-rigid registrering, grey level cooccurrence matrices, Visible Human databasen, voxel-similaritetsmål, fysiske modeller, kontinuummodeller, elastiske modeller, viskose fluide modeller, lineær elasticitet, konvolution, elastisk registrering, fluid registrering, knoglevækst, vækstmodeller, virtual reality, operationssimulering, cranio-facial operationssimulering, finite element modeller, kondenser-

ing, lineære matrixsystemer.

Publications During Ph.D.

This section lists the publications and other work, which have been produced during the Ph.D. period.

Int. Publications related to this thesis

- [BN96e] M. Bro-Nielsen and C. Gramkow, *Fast fluid registration of medical images*, accepted for Visualization in Biomedical Computing (VBC'96), 1996

- [BN96d] M. Bro-Nielsen, *Surgery simulation using fast finite elements*, poster, accepted for Visualization in Biomedical Computing (VBC'96), 1996

- [BN96c] M. Bro-Nielsen and S. Cotin, *Real-time volumetric deformable models for surgery simulation using finite elements and condensation*, Computer Graphics Forum, 15(3):57-66 (Eurographics'96), 1996

- [BN96] M. Bro-Nielsen, *Mvox: Interactive 2-4D medical image and graphics visualization software*, proc. Computer Assisted Radiology (CAR'96), pp. 335-338, 1996

- [Cot96] S. Cotin, H. Delingette, M. Bro-Nielsen, N. Ayache, J.M. Clément, V. Tassetti and J. Marescaux, *Geometric and Physical representations for a simulator of hepatic surgery*, proc. Medicine Meets Virtual Reality, 1996

- [BN95b] M. Bro-Nielsen and S. Cotin, *Soft tissue modelling in surgery simulation for prediction of results of craniofacial operations & steps towards virtual reality training systems*, abstract, proc. 3rd Int. Workshop on Rapid Prototyping in Medicine & Computer-Assisted Surgery, 1995

- [BN95] M. Bro-Nielsen, *Modelling elasticity in solids using Active Cubes - Application to simulated operations*, proc. Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), pp. 535-541, 1995

Int. Publications not related to this thesis

- L. Bjoerndal, T. Darvann, M. Bro-Nielsen and A. Thylstrup, *Automated image analysis applied on odontoblast reactions to caries*, abstract, accepted for Journal of Caries Research, 1996

- T. Darvann, M. Bro-Nielsen, K. Damgaard, U. Knigge, P. Larsen, and L.B. Svendsen, *A possible concept for an interactive 3D visualization system for training and planning of liver surgery*, proc. Medical Informatics Europe (MIE'96), pp. 1052-1054, 1996

- M. Bro-Nielsen, P. Larsen and S. Kreiborg, *Virtual teeth: A 3D method for editing and visualizing small structures in CT scans*, proc. Computer Assisted Radiology (CAR'96), pp. 921-924, 1996

- S. Kreiborg, P. Larsen, M. Bro-Nielsen and T. Darvann, *A 3-dimensional analysis of tooth formation and eruption in a case of Apert syndrome*, poster, proc. Computer Assisted Radiology (CAR'96), pp. 1066-1068, 1996

- B. Nikkhade-Dehkordi, M. Bro-Nielsen, T. Darvann, C. Gramkow, N. Egund and K. Hermann, *3D reconstruction of the femoral bone using two X-ray images from orthogonal views*, poster, proc. Computer Assisted Radiology (CAR'96), pp. 1015, 1996

- M. Bro-Nielsen, K. Suzuki and M. Watanabe, *3D Modeling Method from Occluding Contours by Geometric Primitives*, proc. Asian Conf. Computer Vision (ACCV'93), pp. 221-225, 1993

- K. Suzuki, T. Wada and M. Bro-Nielsen, *Automatic Object Modeling based on Multiview Sensing Images*, proc. Asian Conf. Computer Vision (ACCV'93), pp. 244-247, 1993

Technical reports

- M. Bro-Nielsen, *3D models from occluding contours using geometric primitives*, IMM Tech. Rep. 94-15, 1994

- M. Bro-Nielsen, *Modelling elasticity in solids using active cubes - application to simulated operations*, IMM Tech. Rep. 94-14, 1994

- M. Bro-Nielsen, *Active Nets and Cubes*, IMM Tech. Rep. 94-13, 1994

- M. Bro-Nielsen, *Parallel implementation of Active Nets*, IMM Tech. Rep., 1994

Supervised M.Sc. theses

- [Ras96] Bo Rasmussen, *Craniofacial surgery simulation*, 1996
- Michael Konstantinidis, *Registration and segmentation of medical images*, 1996
- [Gra96] Claus Gramkow, *Registration of 2D and 3D medical images*, 1996
- Rolf Jackson, *On the analysis of 3D images*, 1995

Mvox software package

A software package for visualization and segmentation of medical images has been developed during the Ph.D. [BN96, BN96b]. The software, which is called Mvox, has been commercialized and is being sold by the company Mware. It was displayed at the Computer Assisted Radiology (CAR'96) symposium 1996, where it received a very positive response. Appendix A gives a brief introduction to Mvox. For additional information, please refer to the Mvox WWW homepage [BN96b].

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Medical imaging technologies are altering the nature of many medical professions today. During the last decade many radiology departments have installed powerful imaging equipment for image modalities like Magnetic Resonance (MR), Computed Tomography (CT), and Positron Emission Tomography (PET). These systems have spawned new specialities, and have fundamentally changed the way many diseases are diagnosed, and even the way they are treated.

Smaller CT scanners are being marketed by the imaging system manufacturers, who expect such systems to be increasingly used at clinical department instead of at centralized radiology departments.

In addition, cheap Ultra Sound (US) systems are increasingly being used in the clinical departments. US is experiencing a dramatic development. 3D US is available commercially now, and some people expect the resolution of US to match the present MR resolution, in maybe 5-10 years or less.

Unfortunately, although the 'mechanical' equipment has seen a dramatic development, the real power of these systems has not been released. The software and knowledge, needed to take full advantage of the imaging systems, has not followed the development of the hardware.

It is, therefore, still not unusual that hospitals, even with powerful and expensive 3D scanners, only have diagnostic procedures for handling 2D image slices. In addition, when hospitals do use the 3D reconstruction capabilities of the scanners, the available software is often not sufficiently flexible and advanced, to allow real interaction with the 3D image data and provide useful support for diagnosis.

Only advanced medical imaging research laboratories, which have their own software development capability and access to the latest technology through technical partners, are today able to explore more advanced uses of the 3D images produced by the scanners. Typical for these laboratories has been a specific focus on technical research. The groups have participated in large technical advanced research projects, ie. European Union projects, and have built up the necessary technical expertise through these projects.

But, from these research projects, and other research that is being per-

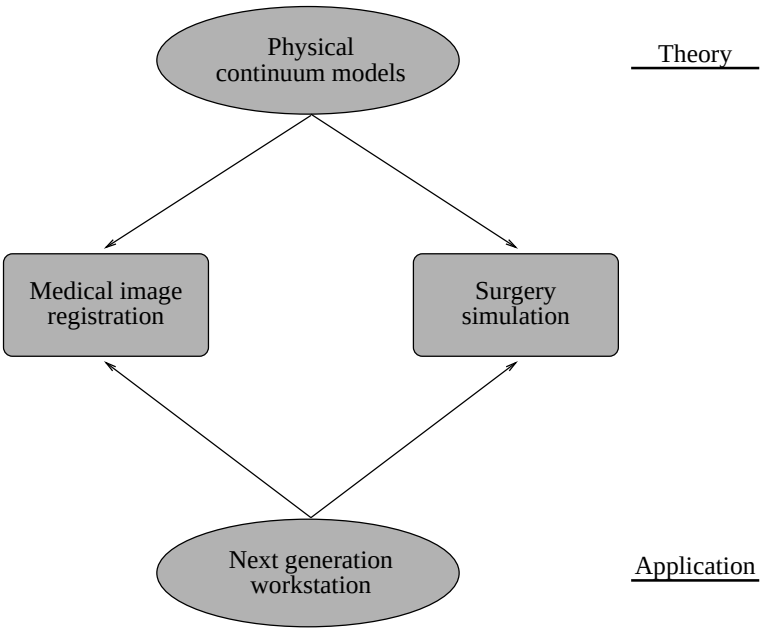

Figure 1.1: *Dual view of medical image registration and surgery simulation.*

formed in the field of medical imaging, new technology is becoming available. Recent years has seen the development of many new image processing, and computer graphics algorithms. Algorithms that ultimately will lead to better medical imaging software for diagnosis, treatment planning, surgery training, and surgery assistance.

This thesis is a part of this volume of work.

The specific subjects of the thesis are Medical Image Registration and Surgery Simulation. In particular, the main focus of the thesis is on the application of physical continuum models of elasticity and viscous fluids in these fields.

The physical models constitute a theoretic view of registration and simulation. A dual application oriented view could also be offered by regarding these technologies as the basic elements of the next generation medical workstation (see figure 1.1).

1.1 Physical models

Physical continuum models are increasingly being used in medical imaging and computer graphics. Both for modeling the natural behaviour of eg. human tissue, but also to control registration processes etc. Unfortunately, fundamental knowledge about their actual physical basis and interrelationships, has been sparse. Models have, therefore, often been used in cases where their assumptions were violated, and numerical problems could be anticipated with sufficient

insight.

This thesis gives a coherent description of the basis of these models, pointing out the algorithmic weaknesses and strengths. In addition, it shows how these models are the backbone of all the image registration and surgery simulation algorithms presented in the thesis.

1.2 Medical workstation of the future

With a longer view, the algorithms and methods, described and developed in this thesis, can be seen as some of the basic elements of the next generation medical workstation. A workstation, which would allow surgeons and other medical staff to diagnose patients, prepare surgery, practice surgery on a virtual patient in the workstation, and perform surgery under the guidance of the computer.

The registration techniques are essential elements of the diagnostic part of the workstation, providing the ability to combine multimodality images for multispectral classification, and automatically segment images using atlas registration. In addition, the physically based methods for surgery simulation are the main technology for creating realistic virtual surgery environments.

1.3 Thesis overview

The main structure of the thesis puts the work on medical image registration first, followed by the work on surgery simulation. Introductions to these fields and reviews are given in chapters 2 and 7 respectively.

The main theoretical chapter of the thesis is chapter 4, where the physical continuum elastic and viscous fluid models are introduced. This chapter can be seen as the hinge of the thesis, around which most of the remaining chapters develop.

The medical image registration work is described in two chapters.

In chapter 3, voxel similarity measures for rigid image registration are evaluated. This chapter discusses the previous work, and shows that almost all of it can be described using Grey Level Cooccurrence Matrices (GLCM). GLCMs are used in texture analysis, and a range of measures based on these matrices exists. A selection of these measures are compared to the existing voxel similarity measures to evaluate them for rigid medical image registration. Images from the Visible Human dataset are used for the evaluation.

After the introduction of the physical continuum models in chapter 4, chapter 5 discusses their application for non-rigid registration of medical images. In particular, a new and faster algorithm for viscous fluid registration is developed.

Chapter 6 presents an alternative approach to non-rigid registration of medical images. It uses actual medical knowledge of the development between two images, to register the images non-rigidly. This chapter can be seen as a response to those people, who have criticized the use of elastic and fluid models in non-rigid medical image registration. They have, correctly, pointed out that

the elastic and fluid models are poor models of the actual physical difference between images. By implementing a computer model of the growth process of the mandibular bone, this chapter shows that it is possible to register two images using a correct physical model.

After the introduction to surgery simulation in chapter 7, chapter 8 describes the development of real-time deformable models for surgery simulation using finite element models of linear elasticity. New ways of improving the real-time response of these models are presented, resulting in socalled Fast Finite Element (FFE) models.

1.4 Thesis contributions

The main contributions of this thesis are listed in this section. Since definitions and nomenclature defined in subsequent chapters are used to keep a compact format, the reader might want to delay reading this section initially.

The main contributions of this thesis are (in order of importance):

Fast Finite Element models These models are developed in chapter 8, and allow real-time deformation of volumetric deformable models. There are three main contributions to the use of finite elements:

1. The use of *condensation* to reduce the complexity of the volumetric models to a complexity similar to that of a surface model (but retaining the volumetric behaviour).
2. The direct *inversion* of the stiffness matrix of the linear matrix equation resulting from the finite element model. This reduces the simulation time dramatically.
3. The use of *static simulation* instead of dynamic simulation. By exploiting the sparseness of the static force vector, a considerable reduction in computation time is achieved.

This work has been published in [BN95b, BN96c, BN96d].

Fast viscous fluid registration algorithm In chapter 5 a new core algorithm for the viscous fluid registration algorithm of Christensen et al. [Chr93, Chr94b, Chr94, Chr96] is developed. This new algorithm allows a speedup of the viscous fluid registration by at least a factor of magnitude. In practice, it means that the registration can be performed on a single workstation instead of on a massively parallel computer.

In addition, this chapter demonstrates that the 'demon'-based registration algorithm of Thirion [Thi96] is the rough simplification of the fluid algorithm.

This work will be published in [BN96e].

Growth-based non-rigid registration Chapter 6 presents a new form of non-rigid registration of medical images. Using the simplicity of the

growth processes of the mandibular bone, a registration algorithm for time sequence images of the mandible is developed. This algorithm uses the actual physical process to control the deformation from one image to the next.

Using the actual physical process to regularize the registration process, rather than an arbitrary elastic or fluid model, has not been described before.

In addition, implicitly, the registration process is also the first computer model of the physical bone growth processes of the human mandible.

This work is being submitted.

Rigid registration using GLCM features By using the analogy with Grey Level Cooccurrence Matrices (GLCM), features from texture analysis are identified that could be used as voxel similarity measures for rigid image registration.

Several of these measures are evaluated for registration and compared to the previously known voxel similarity measures. The experiments are carried out using cryosection, MR, and CT images from the Visible Human data set.

The results are mixed, but show that Mutual Information is the best general voxel similarity measure. For specific modality combinations, some of the new texture measures perform better, though.

Results of registration of CT and MR images, and cryosection and CT images are shown. Voxel-based registration of cryosection and CT images has not been described before.

In addition, it is pointed out that the common scaling of rotations compared to translations using degrees and millimeters, respectively, is wrong. Experiments are carried out to calculate correct scaling factors.

This work is being submitted.

Description of physical continuum models The review and description in chapter 4, of the physical continuum models used in medical image registration and surgery simulation, is believed to be the first complete review of the theoretical aspects of the continuum models and their application in medical imaging.

1.5 Credit

Some of the work in this thesis has been carried out in collaboration with other people. This section specifies their contributions.

1.5.1 Viscous fluid registration

The viscous fluid registration work, described in chapter 5, was initiated during my stay in the Epidaure group of INRIA. It was inspired by the work by Nielsen et al. [Nie94], who showed that Thikonov regularization of a field could be achieved by the simple application of a Gaussian filter, and the work by Thirion [Thi96] on 'demon'-based non-rigid registration. The latter work also used the Gaussian filter.

I continued this work after returning to Denmark. On my initiative and under my supervision, M.Sc. Claus Gramkow joined me during his M.Sc. thesis. Most of the software was written by Claus Gramkow.

1.5.2 Fast Finite Elements

During my stay in the Epidaure group I also developed the ideas for the Fast Finite Element models. I had many discussions with Ph.D. student Stephane Cotin on the basics of finite element models. Stephane Cotin implemented the first finite element models.

The three contributions listed in the section on contributions, were developed by myself in Denmark. Although the use of inversion seems to have some similarity with the approach of Cotin et al. [Cot96], it is formulated in a different way.

1.5.3 Growth registration

The work on bone growth models for non-rigid registration, was strongly inspired by discussions with Prof. Sven Kreiborg. His input, in the form of medical knowledge of the growth of the mandibular bone, was essential for the development of the computer models.

Chapter 2

Medical Image Registration

Segmenting medical images have turned out to be a more difficult task than many image processing researchers originally expected. The individual modalities such as x-ray Computed Tomography (CT), Magnetic Resonance (MR) imaging, Ultra Sound (US), etc., do not individually provide enough contrast and information to reliably segment all tissue types in images acquired of human patients.

Although much research is still being directed at improving segmentation algorithms, the quality of the image data inherently gives a natural limit to the possible quality of the segmentation results. Only when a limited number of objects or tissue classes are present in the images and the imaging process has been designed to provide good contrast for these classes, is it possible to obtain good results. Contrast is sometimes sufficient in one of the normal modalities, eg. bones in CT scans, but often injection of contrast media is used to extract desired anatomical structures.

In addition, sometimes anatomical structures are made of the same tissue on a microscopic level, and only differ in function at a macroscopic level. Eg. brain tissue is very homogeneous but have very different function, depending on the position in the brain. This difference is not visible in local images of the tissue and can only be determined based on the position of the tissue in the global structure of the brain.

Bones are another good example of tissue where function is determined at a macroscopic level. Although they have very similar response in CT scans, they should optimally be segmented into separate structures corresponding to rib, femur, vertebrae, etc. With traditional classification algorithms it is not possible to solve this segmentation problem.

Because of these problems, much attention has been directed towards registration methods in recent years (see [Bro92] for a general review of image registration techniques). Image registration methods can solve some of the inherent problems of mono-modality images and voxel-based classification algorithms.

Rigid multi-modality registration methods allow images of the same patient, but from different modalities, to be registered. This provides joint information

that is:

Complementary: Each modality provides different information. Eg. x-ray Computed Tomography (CT) provides information about bone and calcifications, whereas Magnetic Resonance (MR) provides complementary information about soft tissues.

Synergistic: The combination of two modalities may provide additional information. Eg. Positron Emission Tomography (PET) and Single Photon Emission Computed Tomography (SPECT) provide functional images of the brain, but have very little information about anatomy. Combining PET or SPECT images with MR images increases the value of the PET and SPECT images, since the precisely imaged brain structures in the MR images can be transferred to the functional images. Function can, consequently, be described in terms of anatomy.

The result is that registered images contain more information in each voxel, thus making the segmentation using standard classification algorithms easier.

Another way of introducing more information is using non-rigid registration methods to combine images from either time studies of the same patient or different patients. The latter method is often used with Positron Emission Tomography (PET) images. These images are quite noisy, and the accumulation of information from several different patients, allows the noise to be suppressed statistically.

The use of registered multi-modality images only improves the local information content. The problem of segmenting anatomical structures with similar characteristics is more complex, and a-priori knowledge of anatomy must be put into the segmentation process. To solve this problem, non-rigid registration of functional and topological atlases to patients has been proposed. By mapping the atlas onto the patient it is possible to transfer function, topology, and other information from the atlas to the individual patient.

Registration of medical images has traditionally been performed using either manual methods or extrinsic markers. Unfortunately, these methods have severe disadvantages in terms of precision and/or patient discomfort (stereotactic frames). Much effort has therefore been put into development of non-invasive retrospective image registration techniques, which are more precise and fully automatic. This thesis contributes to this volume of work, and emphasis is therefore put on retrospective image registration.

In this chapter an overview of the techniques used in medical image registration will be presented. Since very good reviews have been published previously by Brown [Bro92], Maurer and Fitzpatrick [Mau93], and Van den Elsen et al. [Els93], there is no reason to repeat this work here. We will instead try to give the reader an understanding of the basic issues in medical image registration and the technical factors that distinguish different registration methods.

There are basicly two factors that influence the classification of medical image registration methods: The *motion model* that determines what transformations are allowed and the *driving potential* that determines the forces driving

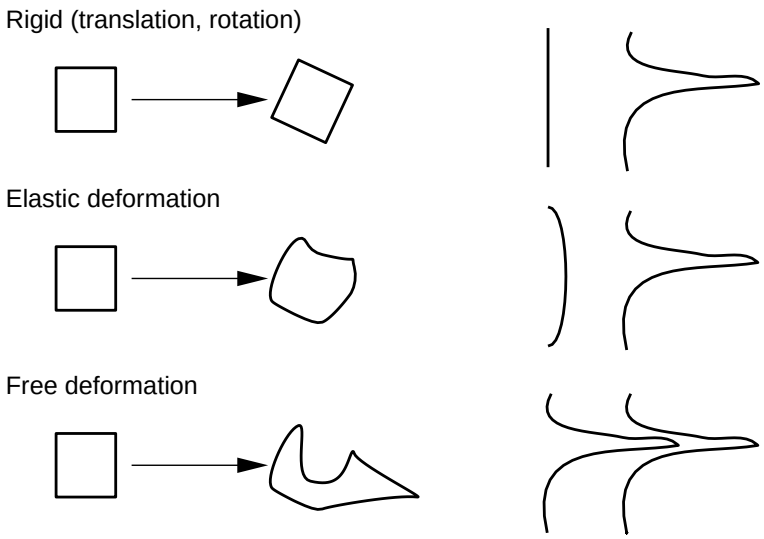

Figure 2.1: *Rigid, elastic, and free motion. Left: Possible deformation of box. Right: Possible deformation of line to curve.*

the motion of the images wrt. each other.

2.1 Motion model

Motion models are normally classified into either *rigid* or *non-rigid* in the computer vision and medical imaging literature. This text proposes an extended classification into three classes: *Rigid*, *elastic*, and *free* motion models. Although elastic and free motion models can be classified together as non-rigid, there are fundamental differences in the resulting transformations they achieve. The three motion models are illustrated in figure 2.1.

2.1.1 Rigid motion

Rigid body motion is composed of a rotation followed by a translation. The body therefore retains its shape and form under a rigid transformation.

Extensions to the basic rigid transformation include scaling and more general affine transformations. These motion models are not used in this thesis and we therefore ignore them.

2.1.2 Elastic motion

Transformations governed by elastic motion models allow constrained deformation of images. The constraints are typicly implemented using potential energy functions for elastic continua, and the transformation becomes a compromise

between the driving forces and the restoring forces of the elastic continuum. Consequently, the driving potential never completely vanishes (see figure 2.1).

Although they never fully achieve the transformation implicitly desired by the driving forces, these models are more robust and smooth than the free motion models, because of the regularization effect induced by the elastic continuum.

In addition to elastic transformations this group of motions also includes polynomial models. These are not used in this work.

2.1.3 Free motion

Free motion models typically allow any deformation that is well-formed. An example of a free motion model is fluid motion. Using fluid motion any deformation can be attained over time, but the transformation is restrained during the deformation process to prevent a breakdown of the well-formedness requirement.

2.1.4 Parameter space

As the variability of these motion models increases, so does the complexity of the parameter space. Whereas rigid motion models have 6 parameters, regularized elastic models typically have in the hundreds of parameters and fluid models in the thousands or millions of parameters. Naturally, with an increasing number of parameters the computational complexity and time consumption increase. On the other hand with an increasing number of parameters more complex motion is possible.

The classification of the motion models can be seen as a motion model scale using the number of parameters as the scale parameter. A multi-scale approach is often used for elastic and free registration where first a rigid registration is applied, followed by an elastic and possibly a free registration. An advantage of this approach is that it introduces a regularization effect in the registration that makes it more robust.

2.2 Driving potential/force

It is difficult to find a useful classification of the possible driving potentials, since the methods often fit several categories at the same time. This is also seen from the different reviews of medical image registration methods [Bro92, Mau93, Els93]. All use different, though related, categories.

We choose to classify the driving potentials into *point*, *curve*, *surface*, *moments and principal axes*, and *voxel similarity* potentials. These are described next.

2.2.1 Point methods

Point methods are used when corresponding sets of points are available in the two datasets. These points can either be *extrinsic* or *intrinsic*. Extrinsic points are typically external skin markers on the patient or markers on stereotactic frames.

Registration using markers on stereotactic frames is usually very precise. But, the discomfort to the patient, caused by the frame which is screwed into the skull of the patient, can be quite serious. See [Mau93] for more discussion on registration using stereotactic frames and pointers to reviews.

External skin markers on the patient come in many different forms and materials (see [Mau93] for discussion of different marker types). A problem with these markers is movement of the skin. But on the other hand they are relatively easy to locate and can be designed to allow subslice/pixel precision (eg. the V shaped markers used by Van den Elsen [Els91])

Intrinsic points are typically anatomic landmarks, such as bloodvessels bifurcations [Hi91], which are located by the physician. Location of these anatomical points can be quite difficult, and the precision seldom match that of extrinsic markers. The advantage of the intrinsic points is their retrospective nature. Extrinsic points require that registration is planned before acquisition of the images, since markers must be physically placed on the patient, whereas intrinsic points are determined in the image without special requirements for the imaging process. This means that historic data (without extrinsic markers) can be registered using intrinsic points. Hill et al. [Hi91, Hi92, Hi92b] suggested several anatomic structures as intrinsic points.

Work by Thirion et al. [Thi93b] indicates that intrinsic points can be found automatically. They detected socalled *extremal points* based on invariants of differential geometry, and have used these points as automatically determined anatomic registration landmarks.

Rigid registration of corresponding point data-sets can be formulated as a least-squares problem. Several closed-form solutions exists [Mau93]: Singular Value Decomposition (SVD) [Aru87], eigenvalue-eigenvector decomposition of a matrix [Hor88] and dual quaternions [Hor87, Fau86]. Point set matching using dual quaternions has been implemented in the software package Mvox [BN96].

Non-rigid registration of point sets has been used both for registration of two different images and for registration to an atlas. See [Mau93] for more details. One method for non-rigid registration of images using corresponding point sets, models one of the images as a thin-plate spline and uses the corresponding points to deform it. An elegant implementation is given by Bookstein in [Boo89], which has also been implemented in the software package Mvox [BN96].

2.2.2 Curve methods

Curves derived from intrinsic structures can be used to register images, if these structures are present and positioned indentically in both images.

Interesting work on this subject has been pursued at the Epidaure group

of INRIA. Based on extrema of curvature in the image intensity field, they extract characteristic geometric features [Mon92, Thi92, Thi93] which they call the *extremal mesh* [Thi93b]. This mesh includes *ridge/crest* lines and *extremal points*.

Gueziec and Ayache smoothed the ridge lines and used a B-spline representation of them to register two CT images rigidly. Thirion et al. used the crest lines for rigid registration of CT images in [Thi92b]. Declerck et al. [Dec95] and Subsol et al. [Sub95] have used the crest lines as stable features of skull anatomy and registered CT images non-rigidly to atlases. Medical applications of this technique were represented in [Sub96].

The limitation of the algorithm seems to be the necessity for high resolution images for stable extraction of the geometric features. But, work is ongoing by Fidrich and Thirion [Fid94] to establish the behaviour of the extremal mesh in scale-space, ie. lower resolution.

In addition, Pennec (eg. [Pen96]) is working on statistical treatment of the geometric features. Crest lines and extremal points are well-suited for statistical analysis of shape, and the application of these geometric features to probabilistic atlases is an interesting aspect.

2.2.3 Surface methods

Three popular registration methods have dominated rigid registration algorithms using surface information: The head-hat algorithm by Pelizarri et al. [Che88, Pel89, Lev88], the Hierarchical Chamfer Matching (HCM) algorithm, which were first proposed by Borgefors [Bor88] based on initial work by Barrow et al. [Brw77], and later used for rigid registration of 3D medical images by Jiang et al. [Jia92, Jia92b], and the Iterated Closest Points (ICP) algorithm by Besl and Kay [Bes92].

The *head-hat* algorithm was developed specificly for registration of images of the head [Che88, Pel89, Lev88]. The surface of one of the images is used as the head and a set of points is extracted from surface contours in the other image to represent the hat. The hat is then registered to the head by minimizing the distance of hat points from the head surface along a line from the point to the centroid of the head surface. The rigid transformation parameters are determined by minimizing the distance energy using Powell's algorithm [Pow64].

In the *Hierarchical Chamfer Matching* (HCM) algorithm, a chamfer distance map (thus the name) is generated from the surface of one of the images [Bor88, Jia92, Jia92b]. This distance map is used as a potential function for surface points in the other image and the total potential is minimized to find the rigid registration parameters.

Instead of using the surface of the image objects, Van den Elsen et al. [Els92] proposed using geometric features extracted from the images using differential geometry. Van den Elsen et al. applied this to registration of MR and CT images.

The *Iterated Closest Points* (ICP) algorithm was introduced by Besl and Kay [Bes92], and later algorithmicly improved by Zhang [Zha94] by the use

of *k-D trees*, as suggested for future work by Besl and Kay. Images must be represented using points on the surface for this algorithm. In each iteration of the algorithm, the closest point in one image is determined for all points in the other image. These point correspondences are used to register the images using closed-form point registration methods as described in section 2.2.1.

The Epidaure group at INRIA has used this algorithm extensively for rigid, affine and locally affine registration (Feldmar and Ayache [Fel94]), and rigid, affine and local spline registration (Declerk et al. [Dec96]). In addition Collignon et al. was inspired by the algorithm in [Col93b].

Rigid registration of 3D surfaces using the ICP algorithm has been implemented in the software package Mvox [BN96].

Szeliski and Lavallee have suggested an alternative representation of the surfaces and their deformation [Sze94]. Using an octree-approximation of the distance potential field and an octree-spline deformation field they registered surfaces non-rigidly.

Other non-rigid surface registration work include that of Moshfeghi et al., who have extended the 2D work by Burr [Bur81] to 3D for non-rigid registration of surface contours [Mos94].

2.2.4 Moments and principal axes methods

With the moments and principal axes methods, image objects are modeled as ellipsoidal point distributions. Such distributions can be described using the first and second moments of the point positions. Registration is performed by overlaying the centroids of the objects, and aligning the principal axes, which are determined by the eigen-vectors of the covariance matrix. Scaling can be determined from the eigen-values of the covariance matrix.

These methods have been widely used for registration of medical images [Alp90, Ara92, Fab88, GA86]. Unfortunately, a major limitation of the moments and principal axes methods is the high sensitivity to shape differences. Missing details or pathology can severely distort the registration results. They are, therefore, mostly used as a rough initial registration step, eg. [Baj89, BN96e].

2.2.5 Voxel similarity methods

Voxel similarity methods register images based on all the voxels in the images. They are therefore, generally robust and results can be quite exact. On the other hand, they have a high computational complexity, and it is only in recent years that practical applications of these methods have turned up.

In this thesis the focus of the registration work is on voxel similarity based methods, and reviews of existing work are therefore giving in the respective chapters. See chapters 3 and 5 for rigid and non-rigid registration methods respectively. In addition see the next section on electronic atlases.

2.3 Registration to electronic atlases

An important application of non-rigid registration methods is registration of images to an electronic atlas. Precise registration allows syntactic and semantic information from the atlas to be transferred to individual patient images.

The immediate application is of course automatic segmentation of images, by transferring the topological information, implicitly stored in the atlas, to the patient. Other information that could be transferred include functional, relational, and hierarchical information.

Another application of registration to an atlas is accumulation of data from images of many patients. This is used in functional imaging, where eg. PET activation images from many patients are registered to the same reference frame, to allow statistical treatment of the functional information.

In PET imaging, the Talairach atlas [Tal88] has often been used as such a reference frame. Patient images are mapped to the atlas by piecewise affine transformations [Eva92, Fox85, Ge92, Lem91]. Although the mapping is rough this approach has been widely used.

Other atlases include the pioneering VoxelMan atlas by Hoehne et al. [Hoe92, Hoe92b, Hoe94]. This atlas has set a standard in visualization and manipulation of anatomical data. It includes both the original modalities (CT/MR) as well as functional, topologic and hierarchic information, thus allowing the user literally to point out regions of a 3D brain and ask 'what is that' or 'what is this part of'.

The development of the Visible Human/Woman [Vis96] data set is another milestone in electronic atlases. Although it has not been completely segmented (or registered), this data set covers the entire body with CT, MR, and RGB cryosection images of histological slices. Several groups are currently working on electronic atlases based on this data set, eg. [Ker96, Tie96].

The algorithms for registration of images to an atlas, can be classified based on the level of human intervention in the registration process.

Manual alignment of image to atlas: Fox et al. [Fox85] and Evans et al. [Eva88] registered images manually to an atlas by performing a rigid or affine transformation followed by nonlinear scaling. Greitz et al. [Gre91] also used manual registration using rigid transformations, but followed this by a small set of non-rigid transformations.

Supervised alignment of image to atlas: These methods are all based on registration using landmarks. Bajcsy et al. [Baj88], Pelizzari et al. [Pel89], and Undrill et al. [Und92] registered the images using rigid transformations followed by scaling. Undrill also allowed non-rigid boundary registration. Bookstein et al. [Boo92] and Evans et al. [Eva91] used thin-plate spline models to elasticly warp the images based on the landmark correspondences.

Automatic alignment of image to atlas: Both Bajcsy et al. [Baj89, Gee93] and Collins et al. [Cli92] used cross-correlation to drive the non-rigid registration. Both groups regularized the deformation by using a multiresolution framework, but Bajcsy et al. also modeled the image as a linear elastic object. Bajcsy et al. were the first to introduce physical continuum models to the problem of voxel-based non-rigid image registration.

Instead of using a multiresolution framework to regularize the registration process, Miller et al. [Mil93, Chr94] used a limited set of eigen-functions of the linear elastic model. This was first applied followed by registration with a full linear elastic model.

In later work Christensen et al. have used viscous fluid continuum models instead of the elastic models, to obtain a more flexible registration, and also to avoid mathematical problems associated with the linear elastic model [Chr93, Chr94b, Chr94c, Chr96].

Subsol et al. [Sub95] used crest lines as the driving potential and registered images to a skull atlas using parametric modal elastic models.

The development of these models has proceeded from manual semi-rigid methods over elastic and parametric/polynomial models to the free registration methods defined by fluids. The fluid models have unfortunately not been used by other groups than Christensen et al., because of the previous enormous computational complexity of the solution problem. In this thesis a new algorithm for fluid registration is proposed, which is considerably faster than the original and, therefore, progress can hopefully be expected with these models.

2.4 Summary

This chapter have presented a review of the most important techniques in medical image registration. Image registration is a wide field and many techniques were described. The following chapters will present new work in voxel-based rigid registration, free fluid registration, and free growth-based registration.

Chapter 3

Rigid Registration using Voxel Similarity Measures

In this chapter, we will concentrate on rigid registration of medical images using voxel similarity measures. An advantage of the voxel-based registration methods presented in this chapter, is the fact that they are fully automatic and require no pre-processing of the images, as do eg. the surface registration methods mentioned in the previous chapter.

The main part of the chapter discusses existing and some new voxel similarity measures. Elaborate tests are used to evaluate the different measures and compare them. Finally, a registration algorithm based on voxel similarity measures is described and some results are presented.

3.1 Image data

The algorithms developed in this chapter have been applied to registration of images from the Visible Human data set [Vis96]. From this data set, images of the head from the following modalitites have been used (see figure 3.1):

- MR, Proton Density weighted (PD)
- MR, T1 weighted (T1)
- CT, windowed for bone (CT)
- Red channel of the cryosection colour image (R)

These images were taken from the Research Systems' Visible Human CD.

Using a combination of manual and automatic tools the images were registered to each other to get an initial ground truth. This registration was performed carefully using visual inspection for validation of the results. Unfortunately, during this registration process, the voxel size of the MR images

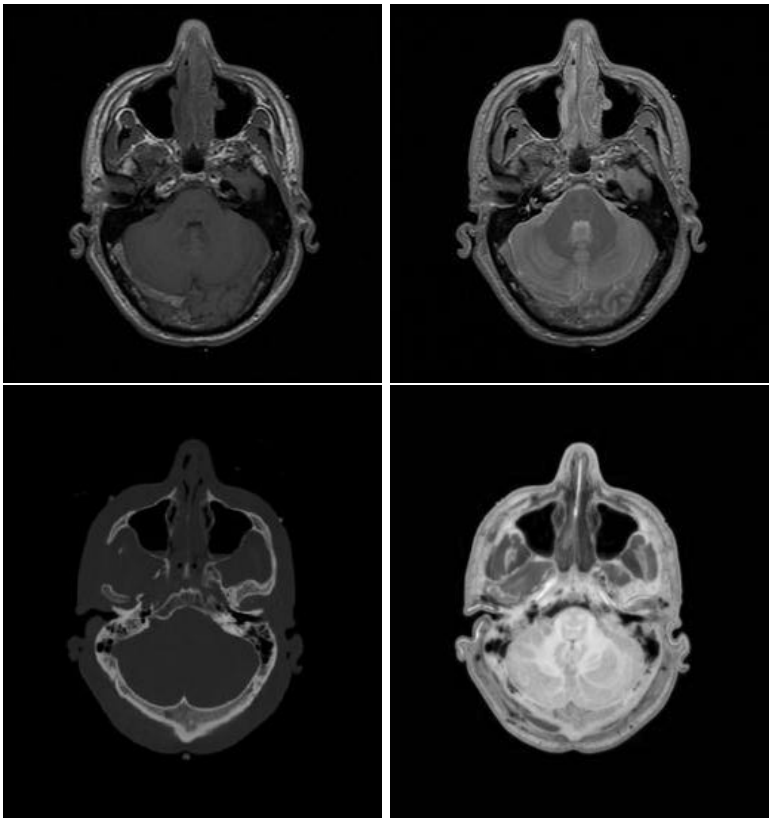

Figure 3.1: *Images from the Visible Human data set. Top left: MR T1 weighted. Top right: MR proton density weighted. Bottom left: CT bone windowed. Bottom right: Red cryosection image.*

turned out to be inconsistent with the size of the other images. By measuring the distance between anatomical landmarks the voxel size were estimated to 1.05 x 1.05 x 5 mm instead of 1.016 x 1.016 x 5 mm as given in the documentation for the MR images.

The following combinations of modalities are explored in this chapter: PD/T1, CT/T1, and CT/R. PD/T1 is only used to show the basic behaviour of the similarity measures and registration results for this combination are consequently not reported. Voxel-based registration of CT and cryosection images has not been documented before.

3.2 Voxel similarity measures

For registration of uni-modal images, correlation has been used extensively in both remote sensing, medical imaging and other application areas.

Simple correlation of grey-values assumes that a *linear* relationship between the grey-values exists [Bro92]. This is seldom the case, and grey-level correlation has, therefore, not provided convincing results for multi-modality registration of images.

In recent years, though, renewed interest in voxel-based multi-modality registration has been revived by the successful work on PET/PET and PET/MR registration by Woods et al. [Woo92, Woo93]. The basic assumption of this work is the same as for correlation, ie. that a linear mapping exists between grey-values g_1 and g_2 of the two images. As mentioned above, this assumption is seldom valid for multi-modality images. But Woods et al. circumvent this problem by looking instead at the variance of the coefficient $R = g_1/g_2$, where g_1 is the PET image grey-value. They argue that this *coefficient of variation* is minimized when the images are in register, and have achieved good results for PET/PET registration [Woo92] using this measure. For PET/MR registration they have proposed a modified version of the initial measure [Woo93], where the variance is calculated independently for each MR grey-value and subsequent summed weighted by the probability estimate of the MR grey-values. To achieve successful registration, only the intracranial structures are used in the registration process, and this algorithm, therefore, needs some manual segmentation to work. But, the coefficient of variation is today probably the best measure for registration of PET/PET and PET/MR [Wes96].

Inspired by this work, Hill et al. proposed a modified algorithm for registration of CT/MR in [Hi93, Hi93b]. In this algorithm CT is used as the denominator g_2, and only certain ranges of CT intensities are used in the calculation of the resulting coefficient.

In [Hi93] Hill also proposed an alternative measure based on the *third order moment* of the 2D histogram created from the images. This was inspired by intensive studies of the development of the 2D histograms for changing registration parameters. A general observation was that intensity concentrations in the histograms seemed to disperse when the registration deviated from an optimal registration.

Van den Elsen has proposed a modified correlation approach for CT/MR registration [Els94, Els94b, Els95], where the images are pre-processed to extract similar structures in both modalities, typicly bones. In [Els94, Els95] these structures were extracted using complex differential operators in scale-space. Similar results were later obtained using simple ramp intensity remapping in [Els94b].

At this point all the measures proposed for multi-modality registration had been based on heuristics. Several groups independently realized that the intrinsic problem of registering two independent image modalities, could be cast in an information theoretic framework. Collignon et al. [Col95] and Studholme et al. [Stu95] both suggested using the *joint entropy* of the combined images as a registration potential, and Collignon et al. [Col95b], and Wells and Viola [Vio95, Wel96] finally suggested the *relative entropy* or *mutual information* as a registration measure. Mutual information is more robust to truncation of images than joint entropy, and has been applied to other registration tasks than medical imaging. It is a very general measure of correspondence between two images, and in a recent evaluation of a range of different multi-modality registration methods [Wes96], mutual information was quite succesful.

It turns out that most of the proposed voxel similarity measures have a correspondence in the field of texture analysis. This chapter shows this correspondence and compares the standard voxel similarity measures to measures used in texture analysis.

3.2.1 GLCM matrices

Except for the work of Van den Elsen [Els94, Els94b, Els95] all the voxel similarity measures introduced above can be formulated based on the 2D histogram or joint probability distribution of the two images.

A similar family of measures is found in the texture analysis literature on *Grey Level Cooccurrence Matrices* (GLCM) [Cnr80, Cnr84, Har73, Har79]. The GLCM is determined as the 2D plot of grey-values of voxels in an image with a fixed displacement between them.

Let $g(\boldsymbol{x})$ be the grey-value of the pixel at position $\boldsymbol{x} = [x_1, x_2, x_3]^T$ in the image, and let $\boldsymbol{u} = [u_1, u_2, u_3]^T$ be the displacement vector between corresponding voxels. The GLCM is generated by accumulating the grey-value pairs $[g(\boldsymbol{x}), g(\boldsymbol{x} + \boldsymbol{u})]$ in a 2D histogram for all image positions $\boldsymbol{x}$. The normalized GLCM can be seen as an estimate of the joint probability distribution of voxels $g(\boldsymbol{x})$ and $g(\boldsymbol{x} + \boldsymbol{u})$.

By extending the definition of the displacement vector $\boldsymbol{u}$ to be, not only between voxels in one image, but also between voxels in *different* images, the GLCM turns out to be the 2D histogram of voxel intensities used by Hill et al. [Hi93, Hi93b, Hi94c], and the normalized GLCM becomes an estimate of the joint probability distribution of voxels in the two images.

In the GLCM texture analysis literature a range of different measures exists. On the following pages we evaluate these measures as voxel similarity measures for multi-modality image registration, and compare them to the existing voxel similarity measures.

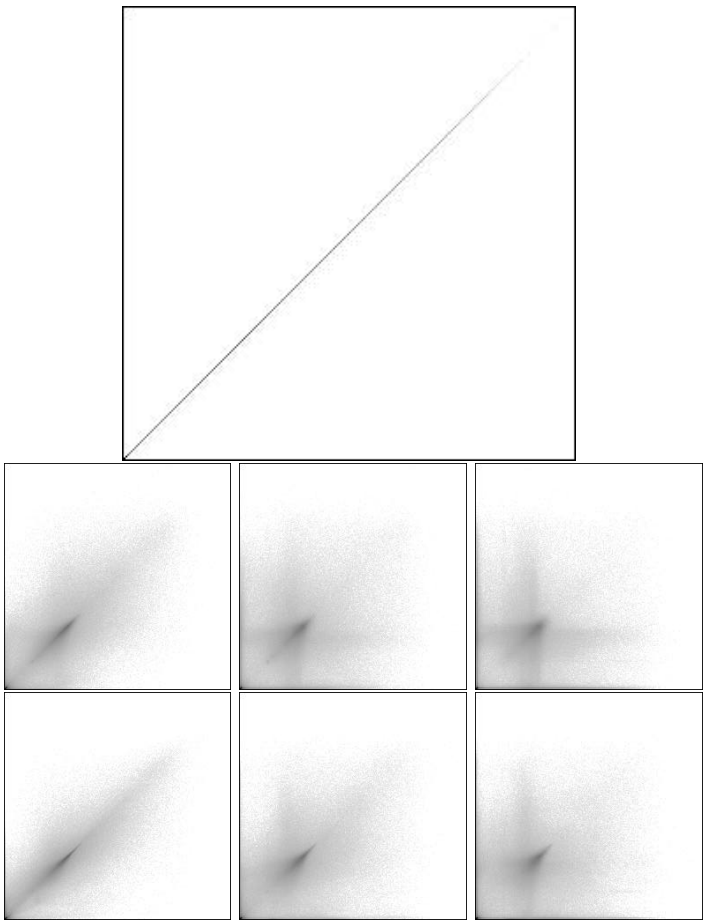

Figure 3.2: *GLCM matrix for MR-T1 against itself. Top: GLCM for registered images. Middle: GLCM when one MR-T1 image is moved 2, 6 and 15 mm in the x-direction. Bottom: GLCM when one MR-T1 image is rotated 2, 6 and 15 degrees around the x-axis.*

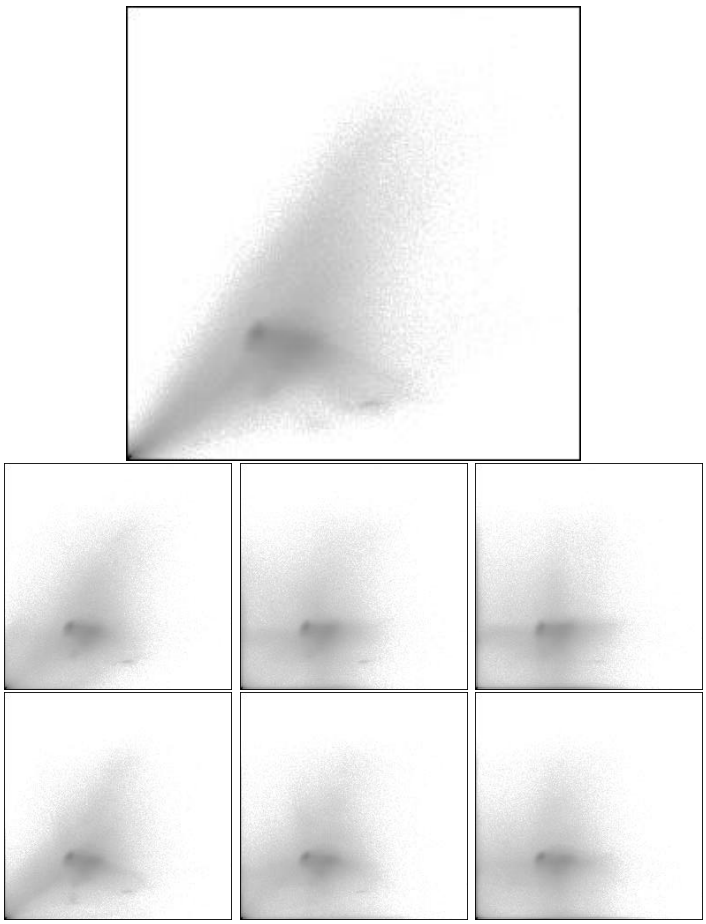

Figure 3.3: *GLCM matrix for MR-PD / MR-T1 images (X/Y). Top: GLCM for registered images. Middle: GLCM when the MR-T1 image is moved 2, 6 and 15 mm in the x-direction. Bottom: GLCM when the MR-T1 image is rotated 2, 6 and 15 degrees around the x-axis.*

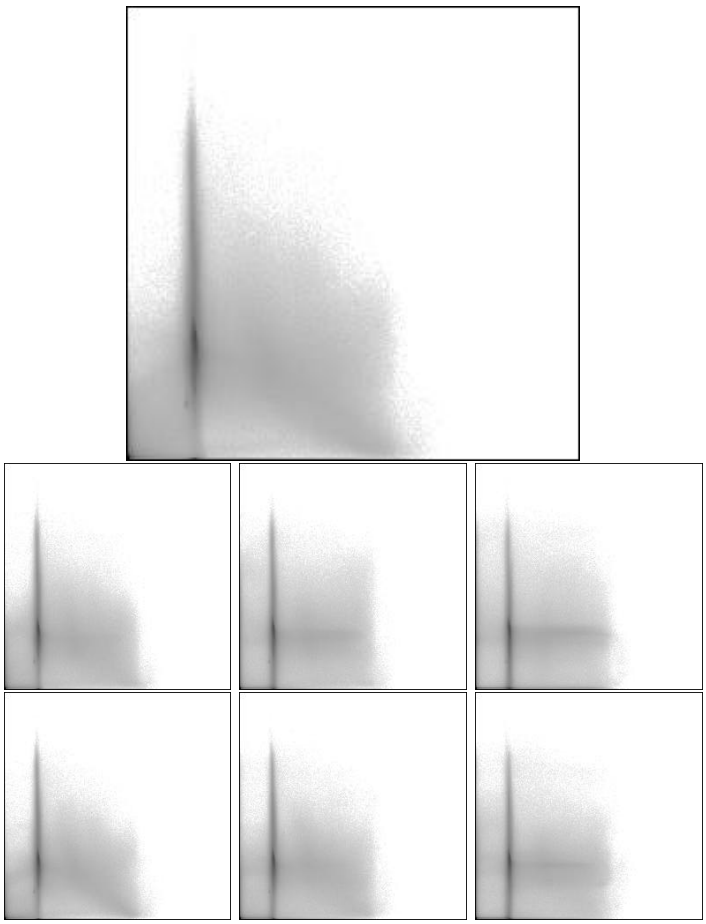

Figure 3.4: *GLCM matrix for CT / MR-T1 (X/Y). Top: GLCM for registered images. Middle: GLCM when the MR-T1 image is moved 2, 6 and 15 mm in the x-direction. Bottom: GLCM when the MR-T1 image is rotated 2, 6 and 15 degrees around the x-axis.*

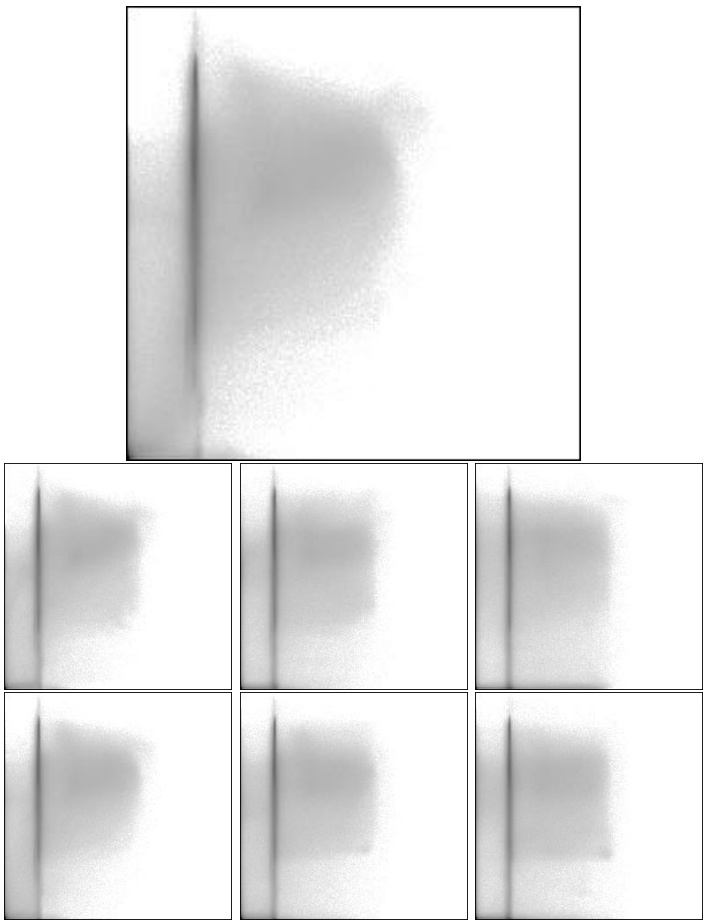

Figure 3.5: *GLCM matrix for CT / Red cryosection (X/Y). Top: GLCM for registered images. Middle: GLCM when the Red image is moved 2, 6 and 15 mm in the x-direction. Bottom: GLCM when the Red image is rotated 2, 6 and 15 degrees around the x-axis.*

The GLCM for two identical images (T1/T1) is shown in the top of figure 3.2. As would be expected, the GLCM shows a diagonal line since identical grey-values have been plotted against each other.

In the lower part of figure 3.2 the evolution of the GLCM is displayed for translation in the x-direction and rotation around the x-axis. The following general observations can be made: With increasing mis-registration,

- diagonal features are dispersed,
- peaks are smoothed out,
- lines parallel to the axes appear.

When the images are in register, grey-values in one image tend to map to a limited range of grey-values in the other image, corresponding to the grey-values of the particular anatomical structures which they both represent. Lines occur since the range of grey-values is contaminated by grey-values from other anatomical structures, when the images are mis-registered. This way each grey-value ends up being mapped to a large range of grey-values in the other image.

Figures 3.3, 3.4 and 3.5 show the characteristics and evolution of the GLCM for the other combinations of modalities. Note, how the GLCMs associated with the CT image have a strong vertical line. This line occurs, because CT has poor contrast for the corresponding tissue type. Since both MR and the red cryosection images have better contrast for this tissue, the CT grey-values are mapped to a wide range of grey-values.

From these GLCMs, the characteristics of the CT/T1 and CT/R registration could be interpreted to be very similar. But this is not the case when features are calculated from the GLCMs and caution should, therefore, be taken when they are interpretated.

3.2.2 GLCM features

In the literature approximately 20 GLCM features appear which can be used to extract information from cooccurrence matrices. Most of these features are derived by weighting the entries of the GLCM with a weighting function and summing the result. Here a limited number of features are investigated. These features fall in three classes based on the character of the weighting function.

Using the notation $P(i,j)$ for elements of the normalized GLCM, the general form of the GLCM features is:

$$F = \sum_{i,j} w(i,j)P(i,j) \tag{3.1}$$

where the weighting function w depends either on the normalized GLCM value $P(i,j)$, the spatial position in the GLCM (i,j), or both:

$$w(i,j) = \begin{cases} f(P(i,j)) \\ f(i,j) \\ f(i,j,P(i,j)) \end{cases} \tag{3.2}$$

Notation

As above $P(i,j)$ is the value of the normalized (n_i, n_j) GLCM at position (i,j).

$$\begin{aligned} N &= n_i n_j \\ P_i(i) &= \sum_j P(i,j) \quad P_j(j) = \sum_i P(i,j) \\ \mu_i &= \sum_i i P_i(i) \quad \mu_j = \sum_j j P_j(j) \\ \sigma_i^2 &= \sum_i (i-\mu_i)^2 P_i(i) \quad \sigma_j^2 = \sum_j (j-\mu_j)^2 P_j(j) \end{aligned} \tag{3.3}$$

$w(i,j)$ is the weighting function applied to the element (i,j) of the normalized GLCM.

Features: Weighting dependent on $P(i,j)$

- **Energy**:

$$\begin{aligned} Energy &= \sum_{i,j} P(i,j)^2 \\ w(i,j) &= P(i,j) \end{aligned} \tag{3.4}$$

- **Variance**:

$$\begin{aligned} Variance &= \sum_{i,j} (P(i,j) - \frac{1}{N})^2 \\ &= (\sum_{i,j} P(i,j)^2) - \frac{1}{N} \\ w(i,j) &\approx P(i,j) \end{aligned} \tag{3.5}$$

- **Entropy**:

$$\begin{aligned} Entropy &= -\sum_{i,j} P(i,j) log P(i,j) \\ w(i,j) &= -log P(i,j) \end{aligned} \tag{3.6}$$

- **Relative entropy (Mutual Information) [Col95b]**:

$$\begin{aligned} MI &= -\sum_{i,j} P(i,j) log \left(\frac{P(i,j)}{P_i(i) P_j(j)} \right) \\ w(i,j) &= -log \left(\frac{P(i,j)}{P_i(i) P_j(j)} \right) \end{aligned} \tag{3.7}$$

Features: Weighting dependent on i, j

- **Inverse Difference Moment**:

$$\begin{aligned} IDM &= \sum_{i,j} \frac{1}{1+(i+j)^2} P(i,j) \\ w(i,j) &= \frac{1}{1+(i+j)^2} \end{aligned} \tag{3.8}$$

- **Inertia**:

$$\begin{aligned} Inertia &= \sum_{i,j} (i-j)^2 P(i,j) \\ w(i,j) &= (i-j)^2 \end{aligned} \tag{3.9}$$

Features: Weighting dependent on $i, j, P(i,j)$

- **Diagonal moment**:

$$\begin{aligned} Dmoment &= \sum_{i,j} |i-j|(i+j-\sigma_i-\sigma_j) P(i,j) \\ w(i,j) &= |i-j|(i+j-\sigma_i-\sigma_j) \end{aligned} \tag{3.10}$$

- **Correlation coefficient**:

$$\begin{aligned} Correlation &= \sum_{i,j} \frac{(i-\mu_i)(j-\mu_j)}{\sigma_i \sigma_j} P(i,j) \\ w(i,j) &= \frac{(i-\mu_i)(j-\mu_j)}{\sqrt{\sigma_i \sigma_j}} \end{aligned} \tag{3.11}$$

- **Cluster shade**:

$$\begin{aligned} Cshade &= \sum_{i,j} (i+j-\sigma_i-\sigma_j)^3 P(i,j) \\ w(i,j) &= (i+j-\sigma_i-\sigma_j)^3 \end{aligned} \tag{3.12}$$

- **Cluster prominence**:

$$\begin{aligned} Cprominence &= \sum_{i,j} (i+j-\sigma_i-\sigma_j)^4 P(i,j) \\ w(i,j) &= (i+j-\sigma_i-\sigma_j)^4 \end{aligned} \tag{3.13}$$

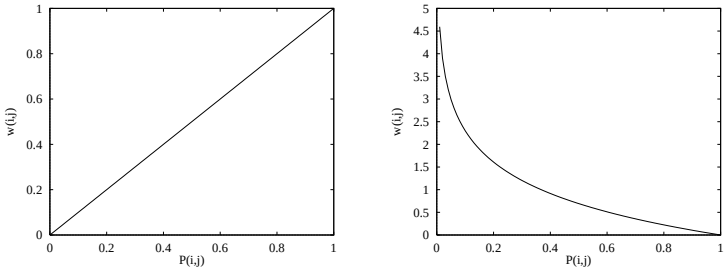

Figure 3.6: *Weighting functions for energy and entropy.*

- **Woods MR-PET [Woo93]**:

$$\begin{aligned} Woods &= \sum_{i>0,j} \frac{s_i}{a_i} P(i,j) \\ w(i,j) &= \frac{s_i}{a_i} \end{aligned} \tag{3.14}$$

where

$$\begin{aligned} a_i &= \frac{1}{P_i(i)} \sum_j P(i,j) j \\ s_i &= \frac{1}{P_i(i)} \sum_j P(i,j)(j-a_i)^2 \end{aligned} \tag{3.15}$$

Note, that the Woods MR/PET registration measure is not symmetric.

Some of these features have been used before for multi-modality image registration (entropy, mutual information, correlation coefficient, and Woods MR/PET) whereas the rest are new for medical image registration. Both groups are included for comparison.

In figure 3.6 the weighting functions for energy and entropy are shown. It is interesting to note that the GLCM images shown previously were pre-processed by applying the log-function. If this is not done, it is impossible to see anything than the largest peaks because of the wide dynamic range of the GLCMs. By applying the log-function, we are able to see the details. The entropy measure, consequently, uses the same weighting function as humans do when we interpretate the GLCM images.

Figure 3.7 shows the weights for the position dependent features. For calculation $\mu_i = \mu_j = 127$, $\sigma_i = \sigma_j = 30$ and the GLCM size is 256×256.

One should keep in mind that the features can be used with an opposite sign and it is, therefore, the *contrast* between different areas of the weight plots that is interesting, not the bright or dark areas alone.

Without going into details, it is clear that the position dependent feature weights cover most general configurations of the GLCM space.

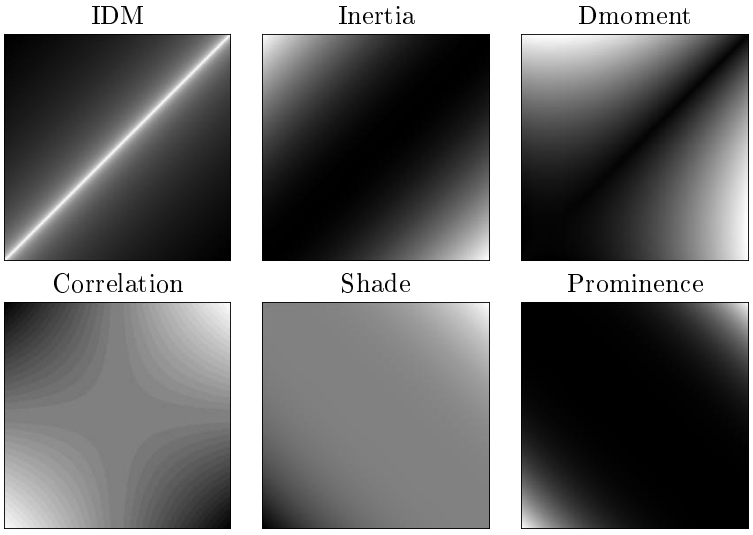

Figure 3.7: *Weights for the position dependent GLCM features. (0,0) is lower left corner and white is high weights.*

3.2.3 Implementation

The rotation of a rigid transformation can be defined in a number of different ways. Two representations are used here: The axes-and-angles system, where the successive rotations around the x-, y- and z-axis are given in angles, and rotation by orthonormal matrices.

The axes-and-angles system is useful for minimization algorithms, since the 6 parameter vector with the 3 rotations and 3 translations is a minimal representation of the rigid transformation. In addition, the axes-and-angles representation gives an intuitive description of the transformation. Results and inputs are therefore given in this format.

The orthonormal matrix representation is useful on the other hand for internal calculations and has been used in all programs. Using orthonormal matrices, a rigid transformation may be written as:

$$\boldsymbol{x}' = \boldsymbol{R}\boldsymbol{x} + \boldsymbol{t} \tag{3.16}$$

where $\boldsymbol{R}$ is a orthonormal rotation matrix and $\boldsymbol{t}$ is a translation vector. In this representation we rotate around the origin of the coordinate system. In practice, we rotate around the center of the image, but we ignore that for simplicity. The rotation matrix is generated from the axis-angles using:

$$\boldsymbol{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} \\ r_{21} & r_{22} & r_{23} \\ r_{31} & r_{32} & r_{33} \end{bmatrix} \tag{3.17}$$

$$
\begin{aligned}
r_{11} &= cos(r_y)cos(r_z) \\
r_{12} &= -cos(r_x)sin(r_z) + sin(r_x)sin(r_y)cos(r_z) \\
r_{13} &= sin(r_x)sin(r_z) + cos(r_x)sin(r_y)cos(r_z) \\
r_{21} &= cos(r_y)sin(r_z) \\
r_{22} &= cos(r_x)cos(r_z) + sin(r_x)sin(r_y)sin(r_z) \\
r_{23} &= -sin(r_x)cos(r_z) + cos(r_x)sin(r_y)sin(r_z) \\
r_{31} &= -sin(r_y) \\
r_{32} &= sin(r_x)cos(r_y) \\
r_{33} &= cos(r_x)cos(r_y)
\end{aligned}
$$

where r_x, r_y and r_z are the rotation angles around the x-, y- and z-axis.

Sample selection

A sample size or sample frequency must be decided upon when the GLCM is calculated. The sample has to be large enough to incorporate enough information about the registration, but at the same time small enough to allow efficient computation.

Collignon et al. [Col95] have illustrated that the sample size influences the behaviour of the similarity measures. They showed that a change from 3 over 1 to 1/3 samples per voxel caused the similarity measures to become increasingly noisy and introduced local minima. In their work [Col95b], they recommend sub-sampling to be used in the beginning, and super-sampling at the end of the iterative process that is used to optimize the registration parameters. Woods et al. [Woo92, Woo93] used a similar scheme, but did not super-sample.

Hill et al. [Stu96b] actually sub-sample the images to create a pyramid using both integer and non-integer reduction factors. They did not super-sample the images.

Wells and Viola [Wel96] used a very different optimization algorithm than did Collignon et al., Woods et al. and Hill et al. In a statistical framework relatively small random samples were used.

Pokrandt [Pok96] has reported preliminary good results using an extremely low 1 in 64 sampling rate combined with random selection of the sampling positions. These results still need to be validated independently.

In this work we have used a scheme similar to that of Collignon et al. The tests described in this section have all been performed using super-sampling with a factor of 2.

Interpolation

When the GLCM is calculated for two images, which overlap in such a way that voxels of one image maps to inter-voxel positions in the other image, it is necessary to estimate the grey-values using interpolation.

Hill et al. and Van den Elsen et al. [Hi94c, Els95] used tri-linear interpolation as the best interpolation method, when computational complexity and quality of the results were weighted.

Collignon et al. [Col95b] recommended *trilinear partial volume distribution* to be used. Instead of using the interpolated grey-value to index the GLCM, they add the partial volume of each of the closest 8 voxels to the GLCM using the grey-values of the original voxels. The advantage should be that no new grey-values are introduced in the images, thus minimizing noise.

For initial runs, nearest neighbour interpolation is recommended in [Col95, Hi94c].

Hill et al. [Hi94b] discusses the use of sinc interpolation and Studholme [Stu96b] comments on using a Gaussian filter, but these interpolation methods have not been used in practice, except in the discussion in [Hi94b].

In this work tri-linear interpolation is used.

Probability estimation

The joint probability $P(i, j)$ is estimated from the GLCM. For 12-bit images the raw GLCM contains 4096×4096 bins which is bigger than some images. A reduction of the number of bins is therefore necessary to allow efficient computation. Parzen-windowing could be used to estimate stable probability estimates [Du73]. But Parzen windowing is computationally expensive. Instead all authours use binning which can be considered a crude approximation of parzen-windowing [Col95b].

In practice the integer grey-values are binarily shifted an appropriate number of bits. Binning is used here.

3.2.4 Plotting GLCM features

In [Hi94b, Hi94c] Hill et al. used socalled *similarity measure plots* to determine the quality of voxel similarity measures. These plots show curves for displacements in the different directions, and rotations around the three axes under the assumption that the other parameters are zero.

Obviously this kind of plot does not provide any information about the quality of the measures for deviations where several parameters are non-zero. In addition, these plots do not allow quantitative evaluation of the measures and objective comparison is not easy. On the other hand they do tend to give a good impression of the behaviour of the measures in terms of local minima and precise localization.

In the following the measures are evaluated using two types of plots:

The similarity measure plot that Hill et al. have used. The similarity measure is determined for a sequence of deviations with a single parameter at a time. This gives a curve for each parameter and these curves are combined in a single plot.

Distance/Feature plots. For a large number of random displacements, the length of the parameter vector is plotted against the feature. It turns out that these plots are reasonably linear for good similarity measures. We therefore choose the linearity as an objective measure of the feature quality. Linear regression [Con84] is used to determine the best approximating line (using the Splus software package) and the R^2 is used as a quality measure.

Correcting for wrong scaling of rotation

When the length of the parameter vector is determined, an implicit choice of scaling for the rotation parameters, compared to the translation parameters, has to be made. The obvious choice is using millimeters for translations and degrees for rotations. In the medical image registration literature this has been used widely (if not exclusively), eg. [Col95b, Els95, Hi94c]. There is no theoretical basis for this choice and any other could just as well have been used. Work in this chapter indicates that it is often wrong.

For algorithms that use a brute-force approach to determine the minimum of the similarity function [Els95, Hi94c] this has little influence. But where more advanced methods such as Newton-Raphson [Woo92, Woo93] or Powell's method [Col95b] are used, different scaling of the rotation and translation axes can influence the direction of steps or stop-requirements. For calculation of the distance/feature plots the scaling also has an effect. It is therefore necessary to estimate the correct scaling.

Two distance/feature plots are created, where one uses only rotation and the other only translation in the parameter vector. Using linear regression, approximating lines are determined for these two plots. Assuming that the estimates of the slopes of the lines are α_{rot} and α_{tr} for rotation and translation, respectively, a correction factor is determined as:

$$\gamma_{rot2tr} = \frac{\alpha_{rot}}{\alpha_{rtr}} \tag{3.18}$$

This correction factor is pre-multiplied all rotation parameters before the length of the parameter vector is determined. Using corrected rotation parameters, a final distance/feature plot is calculated where all parameters take random values.

3.2.5 Results

In appendix D the entire set of plots is shown. The linear regression results for the distance/feature plots are listed along with calculated correction values and R^2 for the corrected and uncorrected plots.

The similarity measure plots call for a subjective evaluation and we have performed this evaluation using the following scale:

1. Useless,

Table 3.1: *Pd-T1: Similarity measure plot of quality result and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	2	0.9384729	0.9088758
Variance	2	0.9384698	0.9088723
Entropy	5	0.9760593	0.9708134
MI	5	0.9260448	0.9214459
IDM	5	0.9685928	0.9599456
Inertia	4	0.9173130	0.9061781
Dmoment	4	0.9187858	0.9159433
Correlation	4	0.9057705	0.8883932
Cshade	3	0.8380936	0.8114711
Cprominence	3	0.8265533	0.8120783
Woods MR/PET (X:Pd)	3	0.7361181	0.7412743
Woods MR/PET (X:T1)	3	0.5480094	0.5482602

2. Poor localization with serious local minima,
3. Reasonable localization of optimum, some small local minima,
4. Reasonable localization of optimum, smooth curve without local minima,
5. Perfect localization of optimum, smooth curve without local minima.

The results of the classification and corrected linear regression are shown in tables 3.1, 3.2 and 3.3. They show that the information theoretic measures entropy and mutual information perform consistently well. This is in line with the image registration literature [Col95, Col95b, Stu95, Stu96, Stu96b] which also indicates that mutual information is better than entropy for truncated images [Stu96], ie. where parts of one image is not present in the other.

The results of the other measures are mixed, but it is interesting to note that the measures with weights based on the position (i, j) (and $P(i, j)$) in the normalized GLCM do quite well in the CT/R experiments. Indeed the Diagonal Moment perform better than the entropy and mutual information measures. Without jumping to any conclusions, this could indicate that position weighted measures can do well if the weighting matches the problem.

3.2.6 Adaptive position-dependent weights

Since the position dependent weights for the feature function can have some positive influence on the resulting behaviour of the feature, it is natural to try to design a position dependent weight function that improves the registration performance.

Work has been done to determine the weights automatically as a function of the correlation coefficient of the individual $P(i, j)$ and the mutual information measure calculated from the images. The idea being that if a position (i, j) is

Table 3.2: *CT/T1: Similarity measure plot quality results and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	4	0.9505296	0.9493070
Variance	4	0.9505302	0.9493080
Entropy	5	0.9666757	0.9638251
MI	4	0.8077108	0.7595025
IDM	2	0.9416978	0.9384330
Inertia	1	0.6917992	0.6948765
Dmoment	1	0.2931991	0.2798332
Correlation	1	0.5524907	0.5277128
Cshade	1	0.0773453	0.0773202
Cprominence	1	0.3667449	0.3952463
Woods MR/PET (X:CT)	4	0.6376055	0.6018653
Woods MR/PET (X:T1)	4	0.8147630	0.8139735

Table 3.3: *CT/R: Similarity measure plot quality results and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	1	0.4335973	0.3803990
Variance	1	0.4335405	0.3803409
Entropy	5	0.9801963	0.9790247
MI	4	0.9016652	0.8648146
IDM	3	0.9476440	0.9375267
Inertia	5	0.9721051	0.9725199
Dmoment	5	0.9896430	0.9720567
Correlation	4	0.9176131	0.8917624
Cshade	4	0.8500580	0.7844821
Cprominence	4	0.8491666	0.7947552
Woods MR/PET (X:R)	3	0.7792640	0.7764421
Woods MR/PET (X:CT)	2	0.9013030	0.8753324

important for the similarity measure, $P(i, j)$ should be strongly correlated with the MI measure.

Unfortunately, it was not possible to obtain better results using the adaptive weight calculation and this work is consequently not reported here.

3.3 Image registration using voxel similarity measures

A registration algorithm similar to that of Collignon et al. [Col95b] has been implemented. The method optimizes the registration using Powell's algorithm for optimization without derivatives [Pow64]. Multi-resolution is used to speedup the algorithm.

A Quasi-Newton algorithm was tested, but problems calculating stable estimates of the first derivatives caused the results to be somewhat poor.

All the voxel similarity measures may be used for the registration. But in practice we have preferred the mutual information most of the time, since it provides consistent results for different modalities.

The result of the 3D registration of the MR T1 weighted image to the CT bone windowed image using mutual information, is shown in figures 3.8, 3.9 and 3.10. The 3D registration of the Red channel of the cryosection image to the CT bone windowed image is shown in figures 3.11, 3.12 and 3.13.

Results of the registration could only be validated by visual inspection and exhaustive test were therefore not performed. But the visual inspection of the results showed that the registration was quite precise.

3.4 Summary

In this section voxel similarity measures for registration of the Visible Human data set have been explored.

The 2D histogram of joint voxel intensities, used in the literature as a basis for definition of many voxel similarity measures, was shown to be similar to the GLCM matrices used in texture analysis of images.

A range of features from texture analysis were compared to the state-of-the-art features. This comparison showed that the state-of-the-art features entropy and mutual information were best for general registration, since they performed consistently well for both registration of MR-T1 to MR-PD, MR-T1 to CT bone, and red cryosection to CT bone. For each of the other combinations, some of the texture measures were at least as good as the information theoretic measures. But, these results were not consistent from one modality combination to the next.

Together with the information from the literature, this leads to the conclusion that mutual information is the best generally applicable voxel similarity measure.

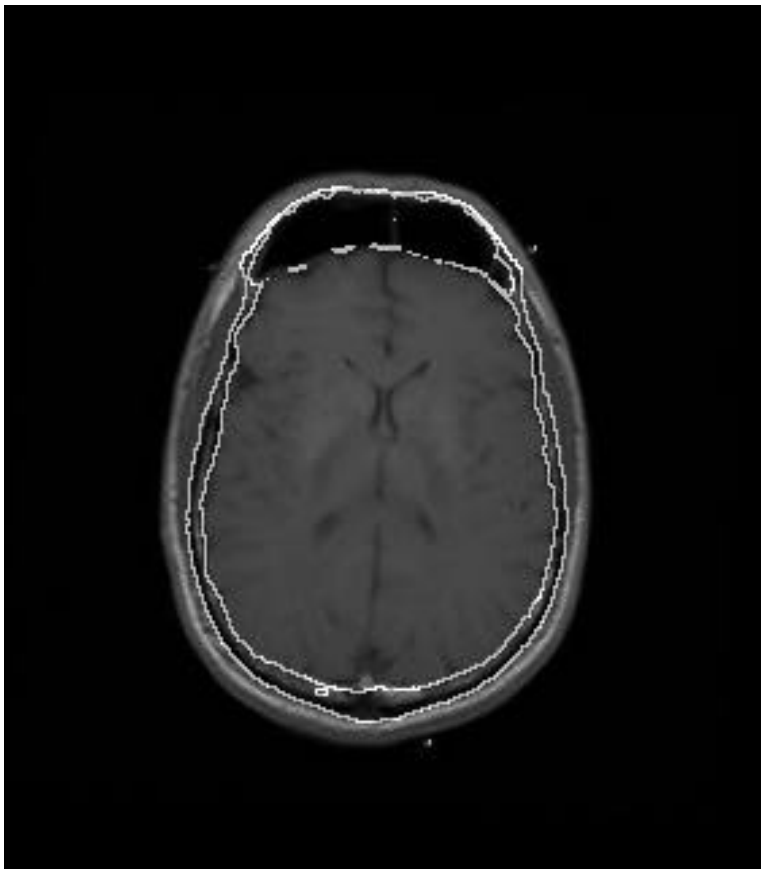

Figure 3.8: *Result of 3D registration using mutual information of the CT bone windowed image to the MR T1 weighted image. The outline of the thresholded CT image has been overlayed on the MR image.*

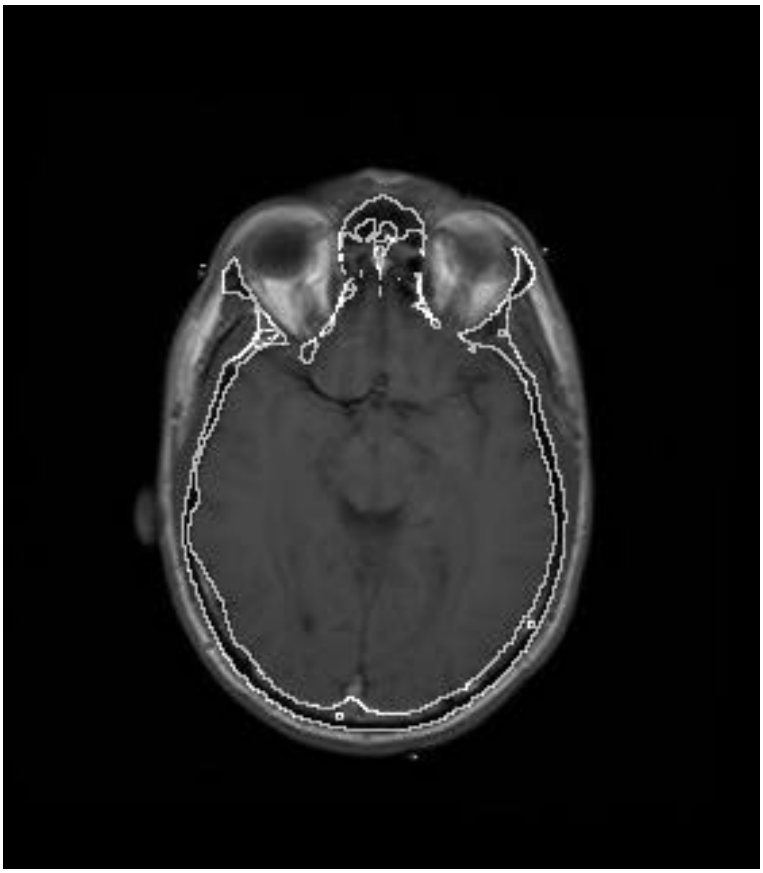

Figure 3.9: *Result of 3D registration using mutual information of the CT bone windowed image to the MR T1 weighted image. The outline of the thresholded CT image has been overlayed on the MR image.*

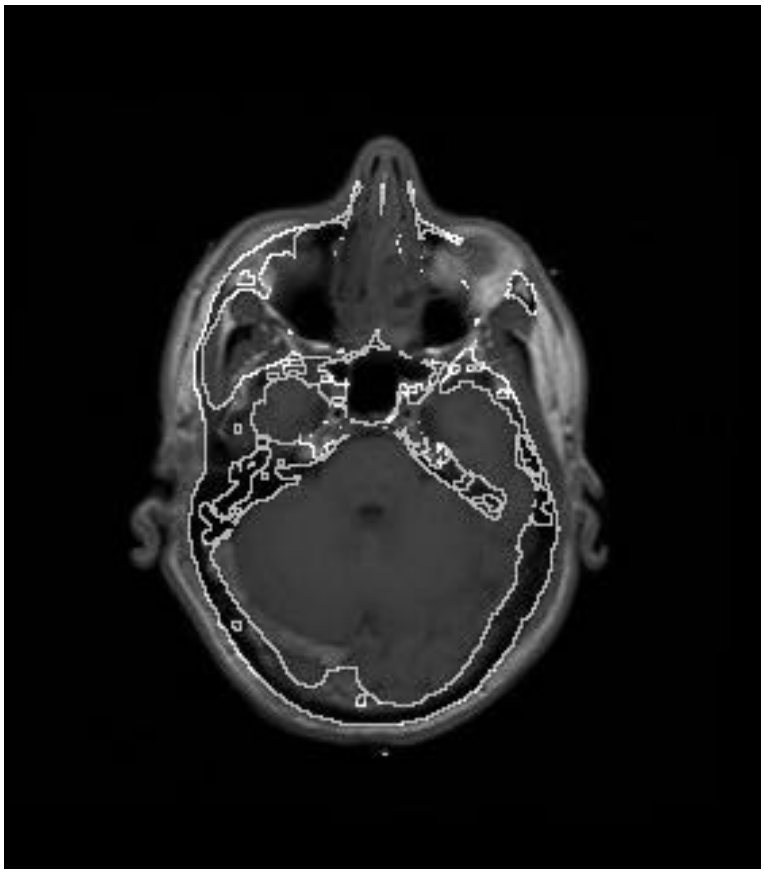

Figure 3.10: *Result of 3D registration using mutual information of the CT bone windowed image to the MR T1 weighted image. The outline of the thresholded CT image has been overlayed on the MR image.*

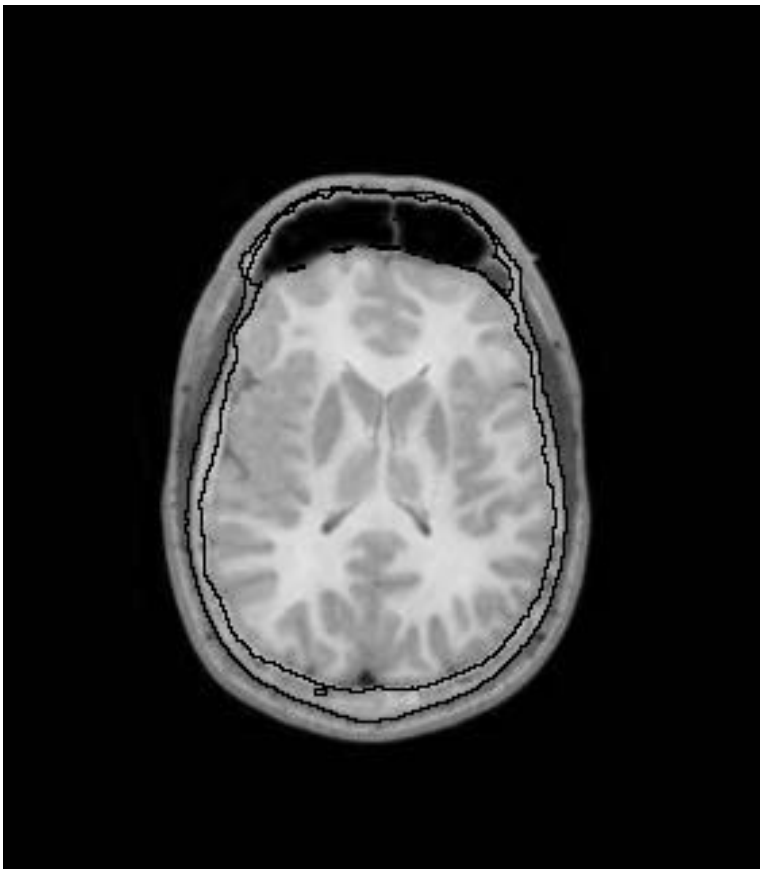

Figure 3.11: *Result of 3D registration using mutual information of the CT bone windowed image to the Red channel of the cryosection image. The outline of the thresholded CT image has been overlayed on the Red image.*

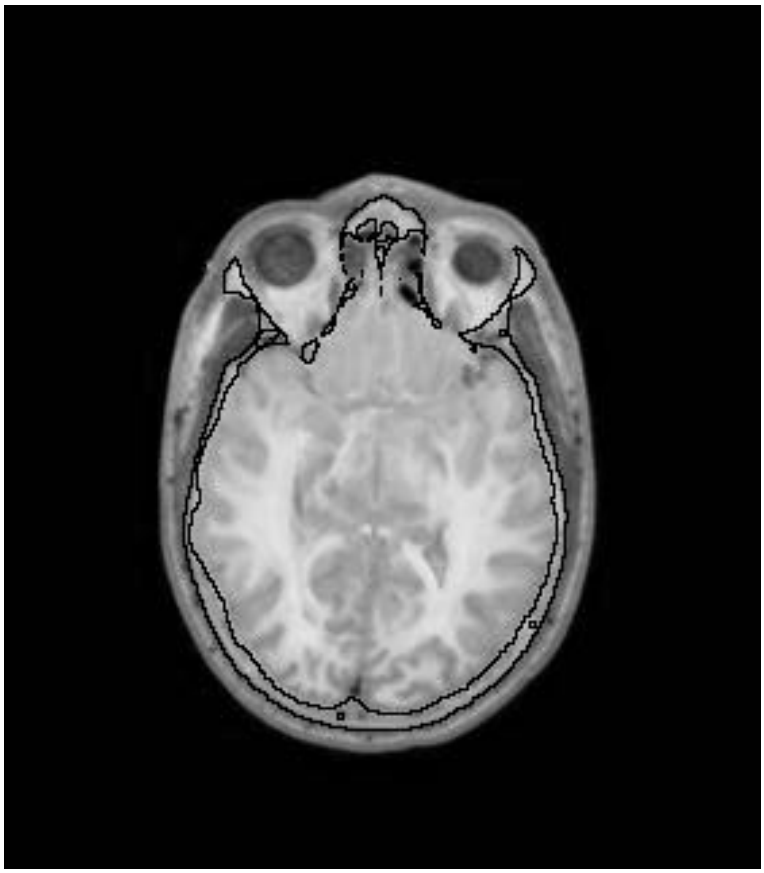

Figure 3.12: *Result of 3D registration using mutual information of the CT bone windowed image to the Red channel of the cryosection image. The outline of the thresholded CT image has been overlayed on the Red image.*

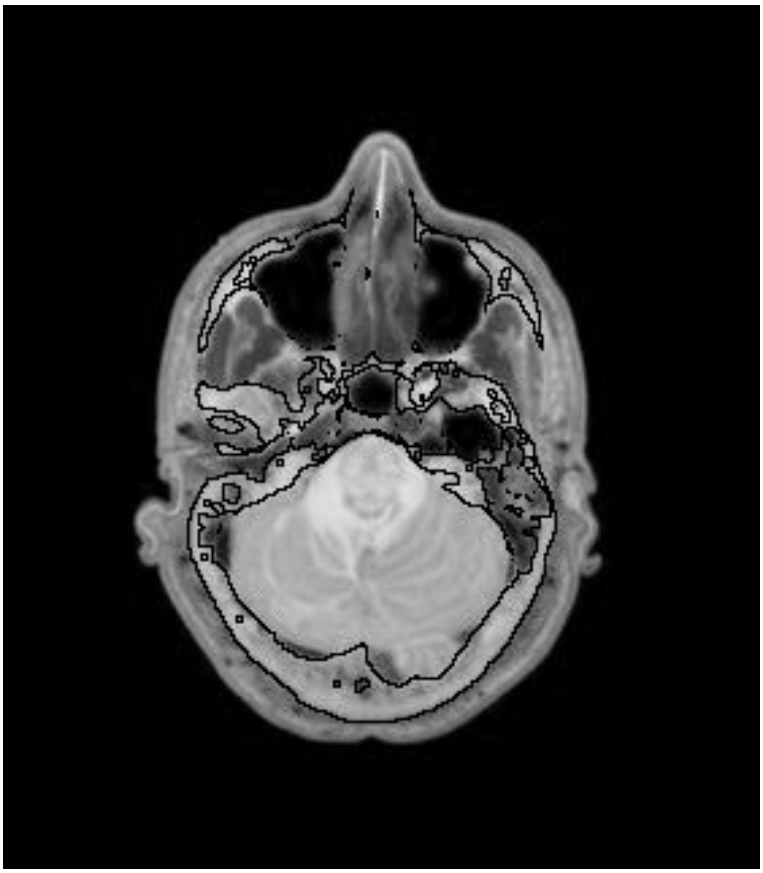

Figure 3.13: *Result of 3D registration using mutual information of the CT bone windowed image to the Red channel of the cryosection image. The outline of the thresholded CT image has been overlayed on the Red image.*

Since most of the texture measures were dependent on the position in the GLCM, in contrast to the information theoretic measures, it should be explored whether position dependent weights adapted to the registration problem (modality combination) could improve registration results. Preliminar work in this direction did not yield positive results.

Chapter 4

Physical Continuum Models

Physical continuum models of elasticity etc. are well-known in computer vision and computer graphics, where they have been used for many years to track and model non-rigid shapes. Although they seldom have been actual *models of the physical phenomena* and rather have been used to *regularize* problems, they have enjoyed great succes.

Because of the emphasis on the regularization effect, these models have often been used without much understanding of their limitations and relationships. But, in recent years, and in particular in the core fields of this thesis - registration and simulation - attention has begun to turn towards understanding rather than blind use of the models.

In medical image registration, some of the first work on non-rigid registration used linear elastic models [Baj89, Chr94]. Christensen et al. illustrated in [Chr94b, Chr94c] that for large deformations, the invalid assumption of linearity of the elasticity operator, can cause changes in the topology of the images. Christensen et al. solved this problem by applying models of fluids [Chr94b, Chr94c] or hyperelasticity (Rabbitt et al. [Rab95]) which can handle large deformations.

This chapter presents a coherent description of the elastic and viscous fluid continuum models of deformation used in the thesis. By using a consistent notation, it is shown how elastic and fluid transformations are closely related to each other. The assumptions used in different practical models are described, and the relationship to previous work is shown.

4.1 Introduction

The continuum models, that are presented in this chapter, can all be described by the sequential relationship between the four factors: Displacement, strain, stress and force (see figure 4.1). By *displacement* we understand the movement of particles in the continuum. This movement changes the configuration of small volume element, and we measure this using *strain tensors* (in practice 3x3

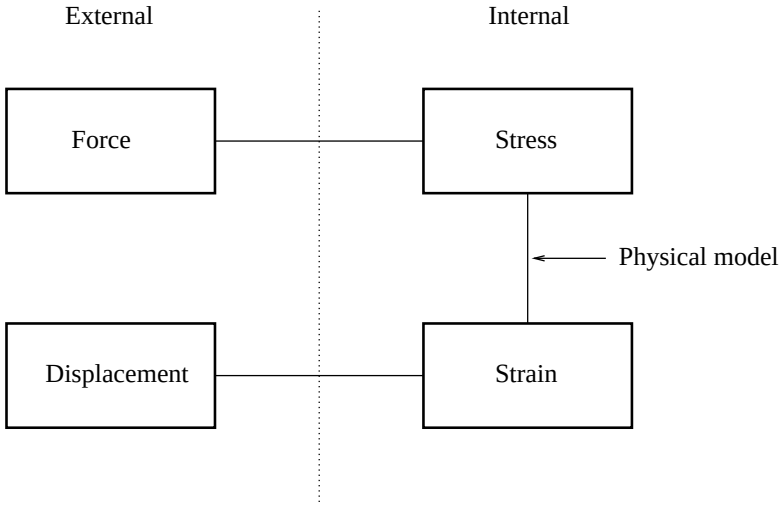

Figure 4.1: *Relationship between physical entities.*

matrices). Using the strain tensors, we can express the elongation, shear etc. of the volume elements.

Forces are similar to displacements, in that they are applied to particles of the continuum. To measure their effect on volume elements we use the *stress tensor*.

Forces and displacements can be seen as external factors that can be observed. Strain and stress are internal mathematical tools to measure the effects of displacements and forces respectively.

The relationship between stress and strain is what determines the actual physical behaviour of the continuum. We study this relationship closer in the sections on elastic and viscous fluid motion, where different models are presented. Although these models have very different practical behaviours, they all satisfy the basic assumptions of continuum mechanics: conservation of mass, force, and momentum.

Indeed, an interesting aspect of this chapter is that the actual physical models are introduced rather late. This illustrates the fact that elastic and viscous fluid models have a common theoretical basis.

Non-linearities occur both between displacement and strain, and strain and stress[1]. Solving the general problems of elasticity and viscous fluid mechanics can, therefore, be quite difficult. To simplify it, some models assume linear relationships. These simplifications are basicly truncated Taylor series, and only valid for small changes in the variables.

We say that a model is a *small displacement/small strain* model for completely linear displacement/stress relationships. When only the strain/stress

[1] Forces are not always constant and can contribute to the non-linearity of the problem. For the simplicity of the presentation we ignore that in this chapter.

relation is linear we say that the model is a *large displacement/small strain* model, etc.

4.2 Mathematical preliminaries

This section presents the mathematical conventions used in this and later chapters.

For sets, the plus operator is defined as working on non-overlapping sets and yielding the union of the sets:

$$A + B \equiv A \cup B, \quad A \cap B = \emptyset \tag{4.1}$$

Vectors and matrices are written with bold letters. Lower case signifies vectors (eg. $\boldsymbol{v}$) and upper case matrices (eg. $\boldsymbol{M}$). In general, the subelements of vectors and matrices are denoted v_i and M_{ij} respectively. In addition, for positions in space we use $\boldsymbol{x} = [x, y, z]^T$.

The zero vector and zero matrix are defined as $\boldsymbol{o}$ and $\boldsymbol{0}$ respectively.

Let $\partial f/\partial x$ be the partial derivative of the function f. We define:

$$f_{x_i} = \partial_i f = \frac{\partial f}{\partial x_i} \tag{4.2}$$

The gradient operator $\boldsymbol{\nabla}$ is defined as:

$$\boldsymbol{\nabla} = \left[\frac{\partial}{\partial x}, \frac{\partial}{\partial y}, \frac{\partial}{\partial z}\right]^T \tag{4.3}$$

Used with a scalar function f it yields the *gradient vector* of the function:

$$\boldsymbol{\nabla} f = [f_x, f_y, f_z]^T \tag{4.4}$$

With vectors we override the $\boldsymbol{\nabla}$ as:

$$\begin{aligned} \boldsymbol{\nabla u} &\equiv (\boldsymbol{\nabla}(\boldsymbol{u}^T))^T \\ &= [\boldsymbol{u}_x, \boldsymbol{u}_y, \boldsymbol{u}_z] = \begin{bmatrix} \partial_1 u_1 & \partial_2 u_1 & \partial_3 u_1 \\ \partial_1 u_2 & \partial_2 u_2 & \partial_3 u_2 \\ \partial_1 u_3 & \partial_2 u_3 & \partial_3 u_3 \end{bmatrix} \end{aligned} \tag{4.5}$$

The divergence of a vector is implemented using $\boldsymbol{\nabla}$ as:

$$\boldsymbol{\nabla} \cdot \boldsymbol{u} \equiv div\ \boldsymbol{u} = \boldsymbol{\nabla}^T \boldsymbol{u} = \partial_1 u_1 + \partial_2 u_2 + \partial_3 u_3 = \mathrm{tr}(\boldsymbol{\nabla u}) \tag{4.6}$$

and for matrices as:

$$\begin{aligned} \boldsymbol{\nabla} \cdot \boldsymbol{M} &\equiv \mathbf{div}\boldsymbol{M} = (\boldsymbol{\nabla}^T \boldsymbol{M})^T \\ &= \begin{bmatrix} \partial_1 M_{11} + \partial_2 M_{21} + \partial_3 M_{31} \\ \partial_1 M_{12} + \partial_2 M_{22} + \partial_3 M_{32} \\ \partial_1 M_{13} + \partial_2 M_{23} + \partial_3 M_{33} \end{bmatrix} \end{aligned} \tag{4.7}$$

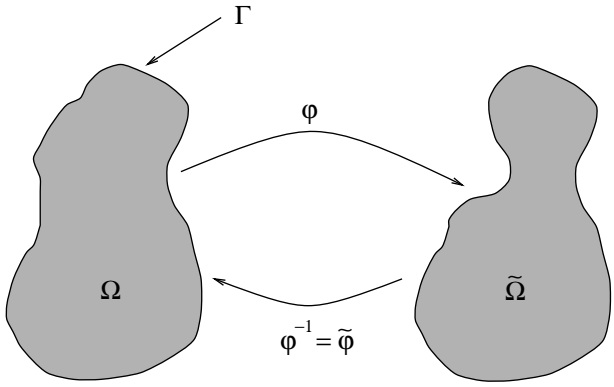

Figure 4.2: *Deformation of the body* $\bar{\Omega}$.

With $\tilde{\boldsymbol{\nabla}}$ we understand the gradient operator wrt. the Eulerian reference frame (introduced later):

$$\tilde{\boldsymbol{\nabla}} = \left[\frac{\partial}{\partial \tilde{x}}, \frac{\partial}{\partial \tilde{y}}, \frac{\partial}{\partial \tilde{z}}\right] \tag{4.8}$$

The derivative wrt. the matrix $\boldsymbol{M}$ is defined as:

$$\frac{\partial \mathrm{W}}{\partial \boldsymbol{M}} = \left[\frac{\partial \mathrm{W}}{\partial M_{ij}}\right] \tag{4.9}$$

The determinant of a matrix $\boldsymbol{M}$ is denoted:

$$|\boldsymbol{M}| = det\boldsymbol{M} \tag{4.10}$$

The *cofactor matrix* $\mathbf{Cof}\,\boldsymbol{M}$ of the matrix $\boldsymbol{M}$ is:

$$\mathbf{Cof}\,\boldsymbol{M} = |\boldsymbol{M}|\boldsymbol{M}^{-T} \tag{4.11}$$

In general we treat tensors as matrices in this text. We do not distinquish between co-variant and contra-variant tensors, although this is abuse of notation. Since we only consider the cartesian coordinate system here, there is no need for this distinction.

4.3 Continuum models

The continuum model, which we present in this section, is the common basis for the elastic and viscous fluid models generally found in the literature and used here.

Part of the notation is superfluous, and some equations are trivial for especially elastic models. But, it is through this insight, that the text attempts to provide an understanding of the commonality of the basic set of notations, equations and axioms.

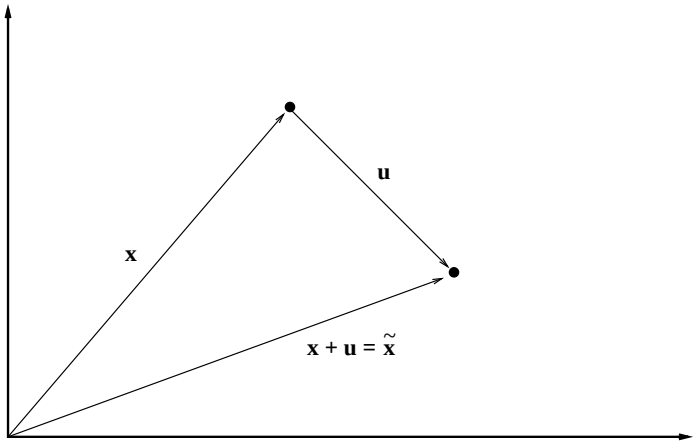

Figure 4.3: *Definition of displacement.*

The presentation builds mostly on [Cia87] for notation and results on elasticity, and on [Car86] for results on fluid mechanics. Rigid proofs are not given here and the reader should consult these texts for additional information.

4.3.1 Definitions

For generality, let us consider the body to be deformed as:

$$\bar{\Omega} = \text{Closure}(\Omega) \tag{4.12}$$

where Ω is an open connected subset of $\mathbb{R}^3$. We call $\bar{\Omega}$ the reference configuration.

By a deformation we will understand a smooth, injective and orientation preserving mapping $\boldsymbol{\varphi}$ [Cia87] that displaces the particle $\boldsymbol{x} = [x_1, x_2, x_3]^T$ to

$$\boldsymbol{\varphi}(\boldsymbol{x}, t) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x}, t) = \tilde{\boldsymbol{x}}(\boldsymbol{x}, t) \tag{4.13}$$

where $\boldsymbol{u}(\boldsymbol{x}, t) = [u_1, u_2, u_3]^T$ is the displacement of particle $\boldsymbol{x}$ at time t. We call $\tilde{\bar{\Omega}} = \boldsymbol{\varphi}(\bar{\Omega})$ the deformed configuration.

Lagrangian and Eulerian reference frames

An alternative formulation of equation 4.13 is:

$$\boldsymbol{x} = \tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t) = \boldsymbol{\varphi}(\tilde{\boldsymbol{x}}, t)^{-1} = \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) \tag{4.14}$$

In this case, the displacement has been parametrized in terms of the deformed variable $\tilde{\boldsymbol{x}}$.

There is an important distinction between the two approaches: In equation 4.13 particles are tracked using their original position as the independent

variable. This parametrization is mostly used in elasticity and is called the *Lagrangian reference frame*. In fluid mechanics, tracking particles using their original position $\boldsymbol{x}$ can be cumbersome for very large deformations. Instead the current position $\tilde{\boldsymbol{x}}$ is used as in equation 4.14. This parametrization is called the *Eulerian reference frame*. We use a tilde ($\tilde{}$) to denote entities parametrized using this reference frame. A particle $\tilde{\boldsymbol{x}}$ in the Eulerian reference frame at time t, originated from position $\boldsymbol{x} = \tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t)$ at time t_0.

Note, that displacements and velocities are physical entities and independent of the reference frame:

$$\begin{aligned} \boldsymbol{u}(\boldsymbol{x}, t) &= \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) \\ \boldsymbol{v}(\boldsymbol{x}, t) &= \tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}, t) \end{aligned} \tag{4.15}$$

In the Lagrangian reference frame the velocities are calculated directly using:

$$\boldsymbol{v}(\boldsymbol{x}, t) = \frac{\partial \boldsymbol{\varphi}(\boldsymbol{x}, t)}{\partial t} \left[= \frac{\partial \tilde{\boldsymbol{x}}(\boldsymbol{x}, t)}{\partial t} \right] = \frac{\partial \boldsymbol{u}(\boldsymbol{x}, t)}{\partial t} \tag{4.16}$$

But, in the Eulerian reference frame the coordinate system depends on the time variable, and a full derivative has to be used:

$$\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}, t) = \frac{d}{dt} \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) \tag{4.17}$$

With the chain rule we find:

$$\frac{d}{dt} \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) = \left. \frac{\partial \tilde{\boldsymbol{u}}}{\partial t} \right|_{(\tilde{\boldsymbol{x}}, t)} + \sum_i \left. \frac{\partial \tilde{\boldsymbol{u}}}{\partial \tilde{x}_i} \right|_{(\tilde{\boldsymbol{x}}, t)} \left. \frac{\partial \tilde{x}_i}{\partial t} \right|_{(\boldsymbol{x}, t)} \tag{4.18}$$

Using equation 4.16 we write this as:

$$\frac{d}{dt} \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) = \left. \frac{\partial \tilde{\boldsymbol{u}}}{\partial t} \right|_{(\tilde{\boldsymbol{x}}, t)} + \tilde{\boldsymbol{\nabla}} \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t) \; \tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}, t) \tag{4.19}$$

which is called the *material derivative* of $\tilde{\boldsymbol{u}}$ in the Eulerian reference frame.

Let us now consider the integral:

$$f = \int_{\tilde{\Omega}(t)} g \; d\tilde{\boldsymbol{x}} \tag{4.20}$$

for the arbitrary function g. We take the material derivative of this integral:

$$\frac{df}{dt} = \frac{d}{dt} \int_{\tilde{\Omega}} g \; d\tilde{\boldsymbol{x}} \tag{4.21}$$

and transform it to the reference configuration ($d\tilde{\boldsymbol{x}} = |\boldsymbol{\nabla} \boldsymbol{\varphi}(\boldsymbol{x})| d\boldsymbol{x}$):

$$\frac{d}{dt} \int_{\tilde{\Omega}} g \; d\tilde{\boldsymbol{x}} = \frac{d}{dt} \int_{\Omega} (g |\boldsymbol{\nabla} \boldsymbol{\varphi}(\boldsymbol{x})|) \; d\boldsymbol{x} \tag{4.22}$$

In this case, since the volume of integration is independent of the time variable t, the derivative can be brought inside the integration sign.

$$\frac{d}{dt}\int_{\tilde{\Omega}} g \; d\tilde{\boldsymbol{x}} = \int_{\Omega} \frac{d}{dt}(g|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|) \; d\boldsymbol{x} \tag{4.23}$$

Using the result that $\partial|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|/\partial t = (\boldsymbol{\nabla}\cdot\boldsymbol{v})|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|$ [Car86] and transforming back to the deformed configuration:

$$\frac{d}{dt}\int_{\tilde{\Omega}} g \; d\tilde{\boldsymbol{x}} = \int_{\tilde{\Omega}} \left[\frac{dg}{dt} + g\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{v}}\right] d\tilde{\boldsymbol{x}} \tag{4.24}$$

Or using the material derivative equation:

$$\frac{d}{dt}\int_{\tilde{\Omega}} g \; d\tilde{\boldsymbol{x}} = \int_{\tilde{\Omega}} \left[\frac{\partial g}{\partial t} + \tilde{\boldsymbol{\nabla}}\cdot(g\tilde{\boldsymbol{v}})\right] d\tilde{\boldsymbol{x}} \tag{4.25}$$

This is the formal statement of the *transport theorem*.

The material derivative and the transport theorem describe how simple variables and integrals can be differentiated wrt. time in the Eulerian reference frame.

4.3.2 Strain

To measure the local deformation we introduce the *deformation gradient*:

$$\boldsymbol{\nabla}\boldsymbol{\varphi} = \left[\boldsymbol{\varphi}_x, \boldsymbol{\varphi}_y, \boldsymbol{\varphi}_z\right] = \begin{bmatrix} \partial_1\varphi_1 & \partial_2\varphi_1 & \partial_3\varphi_1 \\ \partial_1\varphi_2 & \partial_2\varphi_2 & \partial_3\varphi_2 \\ \partial_1\varphi_3 & \partial_2\varphi_3 & \partial_3\varphi_3 \end{bmatrix} \tag{4.26}$$

Since the deformation is orientation preserving by definition, the determinant of the deformation should satisfy the condition:

$$|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x},t)| > 0, \quad \forall \boldsymbol{x} \in \Omega \tag{4.27}$$

$|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x},t)|$ is the Jacobian of the transformation and commonly denoted J in the literature.

We also express the deformation gradient in terms of the *displacement gradient* $\boldsymbol{\nabla}\boldsymbol{u}$:

$$\boldsymbol{\nabla}\boldsymbol{\varphi} = \boldsymbol{I} + \boldsymbol{\nabla}\boldsymbol{u} \tag{4.28}$$

Using the deformation gradient, we define the *right Cauchy-Green strain tensor*[2]:

$$\boldsymbol{C} = \boldsymbol{\nabla}\boldsymbol{\varphi}^T\boldsymbol{\nabla}\boldsymbol{\varphi} \tag{4.29}$$

$\boldsymbol{C}$ is symmetric and positive definite because $\boldsymbol{\nabla}\boldsymbol{\varphi}$ is invertible ($|\boldsymbol{\nabla}\boldsymbol{\varphi}| > 0$).

The strain tensor is supposed to be a measure of the pure deformation of the domain, and independent of rigid transformations. To investigate whether

[2] In differential geometry, this is also known as the *metric tensor*

this is the case, we assume that the deformation $\boldsymbol{\varphi}$ can be split into a non-rigid ($\boldsymbol{u}_d$) and rigid ($\boldsymbol{R}, \boldsymbol{t}$) part:

$$\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{R}(\boldsymbol{x} + \boldsymbol{u}_d) + \boldsymbol{t},\ \boldsymbol{R} \in \mathbb{Q}^3 \tag{4.30}$$

where $\mathbb{Q}^3$ is the set of rotation matrices, ie. orthonormal matrices of size 3 with determinant +1. The deformation gradient becomes:

$$\boldsymbol{\nabla\varphi}(\boldsymbol{x}) = \boldsymbol{R\nabla}(\boldsymbol{x} + \boldsymbol{u}_d) = \boldsymbol{R}(\boldsymbol{I} + \boldsymbol{\nabla u}_d) = \boldsymbol{R\nabla\varphi}_d(\boldsymbol{x}) \tag{4.31}$$

Which also follows from the polar factorization theorem ([Cia87], theorem 3.2-2).

We can now calculate the strain tensor $\boldsymbol{C}$:

$$\begin{aligned} \boldsymbol{C} &= \boldsymbol{\nabla\varphi}(\boldsymbol{x})^T\boldsymbol{\nabla\varphi}(\boldsymbol{x}) = (\boldsymbol{R\nabla\varphi}_d(\boldsymbol{x}))^T\boldsymbol{R\nabla\varphi}_d(\boldsymbol{x}) \\ &= \boldsymbol{\nabla\varphi}_d(\boldsymbol{x})^T\boldsymbol{R}^T\boldsymbol{R\nabla\varphi}_d(\boldsymbol{x}) \\ &= \boldsymbol{\nabla\varphi}_d(\boldsymbol{x})^T\boldsymbol{\nabla\varphi}_d(\boldsymbol{x}) = \boldsymbol{C}_d \end{aligned} \tag{4.32}$$

Thus the strain tensor $\boldsymbol{C}$ is independent of the rigid component. For completely rigid motion the strain tensor is the identity matrix, $\boldsymbol{C} = \boldsymbol{I}$.

Another well-known strain tensor is the *Green-St. Venant strain tensor*[3]:

$$\boldsymbol{E} = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) = \boldsymbol{\nabla u}^T + \boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T\boldsymbol{\nabla u} \tag{4.33}$$

The subtraction of $\boldsymbol{I}$ ensures that this tensor is zero for rigid transformations, and it can, therefore, be seen as a measure of the pure deformation. The factor $\frac{1}{2}$ has been introduced for historical reasons.

In general, a number of different strain tensors exist. There is no 'best' strain tensor, and they can all be derived from each other.

The velocity can be analyzed in a similar fashion. The spatial velocity gradient is defined as:

$$\boldsymbol{L}(\boldsymbol{x}, \boldsymbol{\varphi}) = \boldsymbol{\nabla v}(\boldsymbol{x}) \tag{4.34}$$

The symmetric part of $\boldsymbol{L}$ is called the *rate of strain* tensor:

$$\boldsymbol{D} = \frac{1}{2}(\boldsymbol{L} + \boldsymbol{L}^T) \tag{4.35}$$

4.3.3 Forces

We assume that in the deformed configuration $\tilde{\Omega}$ with the boundary $\tilde{\Gamma} = \partial\tilde{\Omega}$, two kinds of forces are applied to the body:

Applied body forces which work on the interior of the body:

$$\tilde{\boldsymbol{f}} : \tilde{\Omega} \to \mathbb{R}^3 \tag{4.36}$$

where $\tilde{\boldsymbol{f}}$ is the density of the applied body force per unit volume in the *deformed* configuration.

[3] Also known as the *Almansi* or *Green-Lagrange* strain tensor

Applied surface forces which work on the boundary of the domain:

$$\tilde{\boldsymbol{g}} : \tilde{\Gamma}_1 \to \mathbb{R}^3 \tag{4.37}$$

where $\tilde{\boldsymbol{g}}$ is the density of the applied surface forces per unit surface area in the *deformed* configuration, and $\tilde{\Gamma}_1$ is a subset of the full boundary $\tilde{\Gamma} = \tilde{\Gamma}_0 + \tilde{\Gamma}_1$. Fixed displacements are applied to the other part of the boundary ($\tilde{\Gamma}_0$).

We note that these forces are given in the Eulerian reference frame. Since $\boldsymbol{f}(\boldsymbol{x})d\boldsymbol{x} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}})d\tilde{\boldsymbol{x}}$ and $d\tilde{\boldsymbol{x}} = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|d\boldsymbol{x}$ using the Jacobian, we have that:

$$\boldsymbol{f}(\boldsymbol{x}) = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\ \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) \tag{4.38}$$

A similar relation exists for the surface forces [Cia87]:

$$\boldsymbol{g}(\boldsymbol{x}) = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\ \|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})^{-T}\boldsymbol{n}\|\ \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) \tag{4.39}$$

and the mass density

$$\rho(\boldsymbol{x}) = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\ \tilde{\rho}(\tilde{\boldsymbol{x}}) \tag{4.40}$$

4.3.4 Equations of motion for a continuum

The elastic and viscous fluid continuum models all satisfy the basic physical axioms of balance of mass, and linear and angular momentum. We discuss these axioms and their implications in general terms in this section. The next section will introduce model specific considerations.

Conservation of mass

The axiom of conservation of mass, means that the total mass of the body is constant. Constant mass also implies that for an arbitrary volume $\tilde{V} \subset \tilde{\Omega}$ we can write:

$$\frac{d}{dt}\int_{\tilde{V}} \tilde{\rho}(\tilde{\boldsymbol{x}}, t)\ d\tilde{\boldsymbol{x}} = 0 \tag{4.41}$$

where $\tilde{\rho}$ is the density in the deformed configuration.

Setting $g = \tilde{\rho}$ in the transport theorem (4.25) we get:

$$\int_{\tilde{V}} \left[\frac{\partial\tilde{\rho}}{\partial t} + \tilde{\boldsymbol{\nabla}}\cdot(\tilde{\rho}\tilde{\boldsymbol{v}})\right] d\tilde{\boldsymbol{x}} = 0 \tag{4.42}$$

Since the volume $\tilde{V}$ is an arbitrary subset of the domain, we can extract the integrand as the *continuity equation*:

$$\frac{\partial\tilde{\rho}}{\partial t} + \tilde{\boldsymbol{\nabla}}\cdot(\tilde{\rho}\tilde{\boldsymbol{v}}) = 0 \tag{4.43}$$

For incompressible continua, $\tilde{\rho}$ is constant, and:

$$\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{v}} = 0 \tag{4.44}$$

Stress tensor - conservation of momentum

In this section we consider a small volume element. From the axioms of local *balance of linear momentum* and *balance of angular moment* it follows according to the *Cauchy theorem* [Cia87], that for each volume element of the body, there exists a vector field $\tilde{\boldsymbol{t}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{n}})$ and a symmetric tensor field $\tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})$, such that:

$$\tilde{\boldsymbol{t}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{n}}) = \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})\tilde{\boldsymbol{n}} \tag{4.45}$$

where $\tilde{\boldsymbol{n}}$ is the unit outer normal to the surface of the volume element. $\tilde{\boldsymbol{t}}$ and $\tilde{\boldsymbol{T}}$ are called the *Cauchy stress vector* and *Cauchy stress tensor* respectively. At the boundary of the domain $\tilde{\boldsymbol{t}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{n}}) = \tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}})$.

The axiom of *global conservation of linear momentum* can be written for an arbitrary subset $\tilde{V} \subset \tilde{\Omega}$ as:

$$\frac{d}{dt}\int_{\tilde{V}} \tilde{\rho}\tilde{\boldsymbol{v}} d\tilde{\boldsymbol{x}} = \int_{\tilde{V}} \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) d\tilde{\boldsymbol{x}} + \int_{\partial\tilde{V}} \tilde{\boldsymbol{t}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{n}}) d\tilde{\boldsymbol{a}} \tag{4.46}$$

where $\tilde{\boldsymbol{f}}$ is the force working on a volume element in the interior of the body. Using Gauss's divergence theorem, the transport theorem and the continuity equation, we rewrite this as:

$$\int_{\tilde{V}} \tilde{\rho}\left[\frac{\partial\tilde{\boldsymbol{v}}}{\partial t} + (\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{v}})\tilde{\boldsymbol{v}}\right] d\tilde{\boldsymbol{x}} = \int_{\tilde{V}} \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) d\tilde{\boldsymbol{x}} + \int_{\tilde{V}} \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}}) d\tilde{\boldsymbol{x}} \tag{4.47}$$

After collecting terms and using the material derivative equation:

$$\int_{\tilde{V}} \left[\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt} - \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) - \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})\right] d\tilde{\boldsymbol{x}} = 0 \tag{4.48}$$

Since this has to hold for any sub-volume of the continuum, the integrand must be independently zero and we find the *momentum equation*:

$$\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) + \tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{T}} \tag{4.49}$$

Summary

In this section we have derived the general equations of motion for a continuum:

$$\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt} = \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}) + \boldsymbol{\nabla} \cdot \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}}), \quad \tilde{\boldsymbol{x}} \in \tilde{\Omega} \tag{4.50}$$

$$\tilde{\boldsymbol{g}}(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})\tilde{\boldsymbol{n}}, \quad \tilde{\boldsymbol{x}} \in \tilde{\Gamma} \tag{4.51}$$

Together with the formulas for displacement, velocity and strain these equations form a general set of equations for continuum models. In the next section specific models will be introduced linking stress to strain. This linkage is what defines the actual physical characteristics of the continuum.

We note that equations 4.50-4.51 have similar formulations in the Lagrangian reference frame. We associate with $\tilde{\boldsymbol{T}}$ the non-symmetric *first Piola-Kirchoff stress tensor* $\boldsymbol{T}$:

$$\boldsymbol{T}(\boldsymbol{x}) = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})(\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x}))^{-T} = \tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}})\mathbf{Cof}\,\boldsymbol{\varphi}(\boldsymbol{x}) \tag{4.52}$$

The main advantage of this choice is the simple relationship between the divergences of both tensors:

$$\boldsymbol{\nabla}\cdot\boldsymbol{T}(\boldsymbol{x}) = |\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{T}}(\tilde{\boldsymbol{x}}) \tag{4.53}$$

thus giving equation 4.50 in the Lagrangian reference frame as:

$$\left[|\boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})|\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt} =\right] \rho\frac{d^2\boldsymbol{u}}{dt^2} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nabla}\cdot\boldsymbol{T}(\boldsymbol{x}) \tag{4.54}$$

One can also transform the Cauchy stress vector $\tilde{\boldsymbol{t}}$ into a vector in such a way that:

$$\boldsymbol{t}(\boldsymbol{x},\boldsymbol{n}) = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{n} \tag{4.55}$$

where $\boldsymbol{n}$ is the normal vector of the undeformed subdomain. It follows, using equation 4.39, that $\boldsymbol{t}(\boldsymbol{x},\boldsymbol{n}) = \boldsymbol{g}(\boldsymbol{x})$ on Γ_1 and we can write equation 4.51 in the Lagrangian reference frame as:

$$\boldsymbol{g}(\boldsymbol{x}) = \boldsymbol{T}(\boldsymbol{x})\boldsymbol{n}, \quad \boldsymbol{x} \in \Gamma_1 \tag{4.56}$$

In some cases a symmetric stress tensor is desirable. We, therefore, define the *second Piola-Kirchhoff stress tensor* by:

$$\boldsymbol{\Sigma}(\boldsymbol{x}) = \boldsymbol{\nabla}\boldsymbol{\varphi}(\boldsymbol{x})^{-1}\boldsymbol{T}(\boldsymbol{x}) \tag{4.57}$$

For elastic motion, equation 4.54 corresponds to *harmonic motion*. A damping factor $c\frac{d\boldsymbol{u}}{dt}$ is often added to give the equation for *damped harmonic motion*[4]:

$$\rho\frac{d^2\boldsymbol{u}}{dt^2} + c\frac{d\boldsymbol{u}}{dt} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nabla}\cdot\boldsymbol{T}(\boldsymbol{x}) \tag{4.58}$$

For static elastic deformation, the motion equation is rewritten as:

$$\boldsymbol{o} = \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nabla}\cdot\boldsymbol{T}(\boldsymbol{x}) \tag{4.59}$$

The time derivatives are all zero because there is no temporal change.

4.4 Elastic continuum models

An elastic material is characterized by a deterministic relationship between stress and strain which is independent of any history.

[4]The equation for damped harmonic motion is often called the Lagrangian equation of motion

The stress/strain relationship of a general class of elastic materials can be described using the concept of stored energy. These materials are called *hyperelastic materials*, and the *stored energy function* W completely determines the stress/strain relation by:

$$\boldsymbol{\Sigma}(\boldsymbol{x},\boldsymbol{\varphi}) = \boldsymbol{\nabla\varphi}(\boldsymbol{x})^{-1}\boldsymbol{T}(\boldsymbol{x}) = \frac{\partial \mathrm{W}}{\partial \boldsymbol{E}}(\boldsymbol{x},\boldsymbol{\varphi}) \tag{4.60}$$

where $\boldsymbol{E}$ is the Green-St. Venant strain tensor (4.33).

Although these hyperelastic materials do not cover all elastic material models, they form a general class of materials which covers most practical material models.

The potential *strain energy* for the body $\bar{\Omega}$ can be written using the stored energy function as:

$$\mathrm{E}_{strain}(\boldsymbol{\varphi}) = \int_{\Omega} \mathrm{W}(\boldsymbol{x},\boldsymbol{\varphi})d\boldsymbol{x} \tag{4.61}$$

For isotropic and homogeneous materials the stored energy function is completely characterized by the socalled principal invariants of the right Cauchy-Green strain tensor $\boldsymbol{C}$. These invariants appear as the coefficients in the characteristic polynomial $|\boldsymbol{C}-\lambda\boldsymbol{I}|$, and are written in terms of the right Cauchy-Green and Green-St. Venant strain tensors as:

$$\begin{array}{lcll} \iota_1 & = & \mathrm{tr}\boldsymbol{C} & = & 3+2\mathrm{tr}\boldsymbol{E} \\ \iota_2 & = & \frac{1}{2}[(\mathrm{tr}\boldsymbol{C})^2/\mathrm{tr}\boldsymbol{C}^2] & = & 3+4\mathrm{tr}\boldsymbol{E}+2[(\mathrm{tr}\boldsymbol{E})^2-\mathrm{tr}\boldsymbol{E}^2] \\ \iota_3 & = & |\boldsymbol{C}| & = & |2\boldsymbol{E}+\boldsymbol{I}| \end{array} \tag{4.62}$$

Using these invariants the stored energy function for the general hyperelastic material can be formulated as an infinite series:

$$\mathrm{W} = \sum_{rst=0}^{\infty} C_{rst}(\iota_1-3)^r(\iota_2-3)^s(\iota_3-1)^t, \quad C_{000}=0 \tag{4.63}$$

It is interesting to note, that the third invariant measures the volume change of an infinitesimal volume element. For incompressible materials this invariant must be trivially 1. An often used approach to enforce the incompressibility constraint, is to introduce a Lagrange multiplier λ which then works as an internal pressure, forcing the material to find an incompressed equilibrium state. The incompressible version of the stored energy can be written as:

$$\mathrm{W}^i = \sum_{rs=0}^{\infty} C^i_{rs}(\iota_1-3)^r(\iota_2-3)^s + \lambda(\iota_3-1), \quad C^i_{00}=0 \tag{4.64}$$

In the next sections the Mooney-Rivlin and St. Venant Kirchhoff material models are introduced. The *Mooney-Rivlin material model* is a completely non-linear model in both the stress/strain and the displacement/strain relationships. The *St. Venant Kirchhoff material model* is linear in the stress/strain relationship, but non-linear in strain/displacement. Because it is the simplest of the non-linear hyperelastic material models, it is often used. More popular, though, is

the linearized version, where the strain/displacement relationship is made linear by the assumption of infinitesimal displacements. This linearized version of the St. Venant Kirchhoff model is also known as *linear elasticity* or *Hooke's Law*.

4.4.1 Mooney-Rivlin materials

The *compressible* Mooney-Rivlin material model is described by the approximation to the general stored energy function by:

$$\begin{aligned} \mathrm{W} &= C_{100}(\iota_1 - 3) + C_{010}(\iota_2 - 3) + \Gamma(\sqrt{|\iota_3|}) \\ \Gamma(\delta) &= c\delta^2 - d\log\delta, \quad c, d > 0 \end{aligned} \tag{4.65}$$

The Γ-function ensures that large volume changes are accompanied by infinite stress.

The incompressible version ignores the third element and can be written as:

$$\mathrm{W} = C_{100}(\iota_1 - 3) + C_{010}(\iota_2 - 3), \quad |\iota_3| = 1 \tag{4.66}$$

The Mooney-Rivlin material model has been used in surgery simulation and for general modeling of human tissue, eg. [Sag94, Beg88].

4.4.2 St. Venant Kirchhoff materials

For $C_{rst} = 0$ except for

$$C_{100} = \mu \quad C_{200} = \frac{\lambda + 2\mu}{8} \quad C_{010} = -\frac{\mu}{3} \tag{4.67}$$

we get the St. Venant Kirchhoff material model:

$$\mathrm{W} = \frac{\lambda}{2}(\mathrm{tr}\boldsymbol{E})^2 + \mu \mathrm{tr}\boldsymbol{E}^2 \tag{4.68}$$

where μ and λ are the Lamé material constants.

Note, that this material model has a linear stress/strain relationship, since:

$$\boldsymbol{\Sigma} = \frac{\partial \mathrm{W}}{\partial \boldsymbol{E}} = \lambda(\mathrm{tr}\boldsymbol{E})\boldsymbol{I} + 2\mu\boldsymbol{E} \tag{4.69}$$

and that the volume change measured by ι_3 does not play any role.

The St. Venant Kirchhoff material has been used in computer graphics (eg. Terzopoulos [Ter88]).

4.4.3 Linear elastic materials

Linear elasticity is with no doubt the most well-known of the elastic models. It has the remarkable property, that it is completely linear in terms of the force/displacement relationship. Systems derived from the linear elastic operator are, consequently, easy to solve compared to the other elastic models

which in general are non-linear. Unfortunately, the assumption leading to the linerization of the St. Venant Kirchhoff material model is seldom satisfied.

The basic assumption is that the non-linear Green-St. Venant strain tensor:

$$E = \frac{1}{2}(\boldsymbol{C} - \boldsymbol{I}) = \frac{1}{2}[\boldsymbol{\nabla u}^T + \boldsymbol{\nabla u} + \boldsymbol{\nabla u}^T \boldsymbol{\nabla u}] \tag{4.70}$$

can be approximated by the *linearized strain tensor*

$$E_l = \frac{1}{2}[\boldsymbol{\nabla u}^T + \boldsymbol{\nabla u}] \tag{4.71}$$

This is only valid for infinitesimal displacements, since in that case the quadratic term $\boldsymbol{\nabla u}^T \boldsymbol{\nabla u}$ disappears.

Using the linearized Green-St. Venant strain tensor we can write the stored energy function for the linear approximation of the St. Venant Kirchhoff material model as:

$$\mathrm{W}_l = \frac{\lambda}{2}(\mathrm{tr}\boldsymbol{E}_l)^2 + \mu \mathrm{tr}\boldsymbol{E}_l^2 \tag{4.72}$$

The *linear elastic* material model is popularly known as *Hooke's Law*.

To follow the *"engineering"* notation often used with linear elasticity, we introduce the *engineering strain vector* $\boldsymbol{\epsilon}$:

$$\boldsymbol{\epsilon} = [u_x, v_y, w_z, u_y + v_x, u_z + w_x, v_z + w_y]^T \tag{4.73}$$

The linearized strain tensor can be written as:

$$\begin{aligned} \boldsymbol{E}_l &= \frac{1}{2}[\boldsymbol{\nabla u}^T + \boldsymbol{\nabla u}] \\ &= \begin{bmatrix} u_x & \frac{1}{2}[u_y + v_x] & \frac{1}{2}[u_z + w_x] \\ \frac{1}{2}[u_y + v_x] & v_y & \frac{1}{2}[v_z + w_y] \\ \frac{1}{2}[u_z + w_x] & \frac{1}{2}[v_z + w_y] & w_z \end{bmatrix} \end{aligned} \tag{4.74}$$

We note that:

$$\begin{aligned} \mathrm{tr}\boldsymbol{E}_l &= [111000]\boldsymbol{\epsilon} = \boldsymbol{h}^t \boldsymbol{\epsilon} \\ \mathrm{tr}\boldsymbol{E}_l^2 &= \boldsymbol{\epsilon}^T \mathbf{Diag}\left[\mathbf{111}\frac{\mathbf{1}}{\mathbf{2}}\frac{\mathbf{1}}{\mathbf{2}}\frac{\mathbf{1}}{\mathbf{2}}\right]\boldsymbol{\epsilon} = \boldsymbol{\epsilon}^{\mathbf{T}} \boldsymbol{H} \boldsymbol{\epsilon} \end{aligned} \tag{4.75}$$

We can thus rewrite the linearized St. Venant Kirchhoff stored energy function as:

$$\begin{aligned} \mathrm{W}^l &= \frac{\lambda}{2}(\mathrm{tr}\boldsymbol{E}_l)^2 + \mu \mathrm{tr}\boldsymbol{E}_l^2 \\ &= \frac{\lambda}{2}(\boldsymbol{h}^T \boldsymbol{\epsilon})^2 + \mu \boldsymbol{\epsilon}^T \boldsymbol{H} \boldsymbol{\epsilon} \\ &= \frac{1}{2}\boldsymbol{\epsilon}^T [\lambda \boldsymbol{h}\boldsymbol{h}^T + \mu \boldsymbol{H}]\boldsymbol{\epsilon} \\ &= \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{M} \boldsymbol{\epsilon} = \frac{1}{2}\boldsymbol{\epsilon}^T \boldsymbol{\sigma} \end{aligned} \tag{4.76}$$

where $\boldsymbol{\sigma} = \boldsymbol{M}\boldsymbol{\epsilon}$ is the *engineering stress vector* and $\boldsymbol{M}$ is the *material matrix*:

$$\boldsymbol{M} = \lambda \boldsymbol{h}\boldsymbol{h}^T + \mu \boldsymbol{H} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{4.77}$$

To those familiar with linear elasticity, the stored energy function might be more recognizable, when written using the actual partial derivatives:

$$\mathrm{W}^l = \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\lambda}{2} \left(\partial_i u_i\right)\left(\partial_j u_j\right) + \frac{\mu}{4}\left(\partial_j u_i + \partial_i u_j\right)^2 \tag{4.78}$$

The governing partial differential equation (PDE) for static linear elastic deformation is formulated using equation 4.69:

$$\begin{aligned} \boldsymbol{o} &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\nabla}\cdot\boldsymbol{T}(\boldsymbol{x}) \\ &= \boldsymbol{f}(\boldsymbol{x}) + \mu\boldsymbol{\Delta}\boldsymbol{u}(\boldsymbol{x}) + (\mu+\lambda)\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x})) \end{aligned} \tag{4.79}$$

where $\boldsymbol{\Delta} = \boldsymbol{\nabla}^T\boldsymbol{\nabla}$ is the Laplacian operator.

The linear elastic material model has been used in both computer graphics (eg. [Gou89, Pie92, BN96c]), and image processing (eg. [Baj89, Mil93]).

Two other well-known linear material models are the Laplacian or membrane model:

$$\begin{aligned} \mathrm{W}^l_{membrane} &= \sum_{i=1}\left(\partial_i u_i\right)^2 \\ 0 &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\Delta}\boldsymbol{u}(\boldsymbol{x}) \end{aligned} \tag{4.80}$$

and the biharmonic or thin-plate model:

$$\begin{aligned} \mathrm{W}^l_{thin-plate} &= \sum_{i=1}\sum_{j=1}\left(\partial^2_{ij} u_i\right)^2 \\ 0 &= \boldsymbol{f}(\boldsymbol{x}) + \boldsymbol{\Delta}^2\boldsymbol{u}(\boldsymbol{x}) \end{aligned} \tag{4.81}$$

The membrane model can be seen as a simplification of the linear elastic model, where the cross-directional effects due to $\boldsymbol{\nabla}(\boldsymbol{\nabla}\cdot\boldsymbol{u}(\boldsymbol{x}))$ are ignored.

These models have been used extensively in image processing and computer vision, both alone (eg. [Boo89]) and together (eg. [Kas88, Coh89]).

4.5 Viscous fluid continuum models

In contrast to elastic models the strain in fluid models is history dependent. Elastic models are characterized by spatial smoothing of the displacement field.

Fluid models, on the other hand, are characterized by spatial smoothing of the velocity field. In practice, this means that any displacement can be obtained given enough time because the internal stresses in the fluid disappear temporarily.

The fluid model used in this thesis is defined by the viscous fluid Navier-Stokes equations, which we derive below.

As a first step, the viscous and pressure contributions to the stress tensor are written explicitly:

$$\tilde{\boldsymbol{T}} = \tilde{\boldsymbol{V}} - \tilde{p}\boldsymbol{I} \tag{4.82}$$

where $\tilde{\boldsymbol{V}}$ is the reduced stress tensor representing the viscous effects, and $\tilde{p}$ is the pressure.

The basic assumption of the Navier-Stokes equations is based on *Stokes' hypothesis*. This hypothesis states that $\tilde{\boldsymbol{V}}$ is a function of the deformation rate tensor $\tilde{\boldsymbol{D}} = \frac{1}{2}(\tilde{\boldsymbol{L}} + \tilde{\boldsymbol{L}}^T)$ which is the symmetric part of the spatial velocity gradient $\tilde{\boldsymbol{L}}$ (equation 4.34). The relation is:

$$\tilde{\boldsymbol{V}} = 2\mu\tilde{\boldsymbol{D}} + \lambda(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}})\boldsymbol{I} \tag{4.83}$$

Inserting this into the momentum equation gives:

$$\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt} = \tilde{\boldsymbol{f}} - \tilde{\boldsymbol{\nabla}}\tilde{p} + \mu\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}} + (\lambda + \mu)\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}}) \tag{4.84}$$

where $\tilde{\boldsymbol{\Delta}} = \tilde{\boldsymbol{\nabla}}^T\tilde{\boldsymbol{\nabla}}$ is the Laplacian[5].

For very low Reynold's number [6] flow, a simplified model is obtained when we ignore the pressure gradient $\tilde{\boldsymbol{\nabla}}\tilde{p}$ and the inertial term $\tilde{\rho}\frac{d\tilde{\boldsymbol{v}}}{dt}$ [Chr94c]:

$$\boldsymbol{o} = \tilde{\boldsymbol{f}} + \mu\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}} + (\lambda + \mu)\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}}) \tag{4.85}$$

Surprisingly, this is actually the linear elasticity PDE (equation 4.80), where the displacement $\boldsymbol{u}$ has been replaced by the velocity $\tilde{\boldsymbol{v}}$. This similarity is used extensively to transfer results from linear elasticity to the solution of the viscous fluid problem.

4.6 Eigen-function parametrization of the linear elastic operator

In this section we focus on the linear operator associated with linear elasticity:

$$\mathcal{L}\boldsymbol{u} = \mu\boldsymbol{\Delta}\boldsymbol{u}(\boldsymbol{x}) + (\mu + \lambda)\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{u}(\boldsymbol{x})) \tag{4.86}$$

Following the approach of Miller, Christensen, et al. [Chr94, Chr94c, Mil93], we reparametrize the linear elasticity operator using the eigen-functions and eigenvalues for a specific case of boundary conditions. Since mapping the boundary

[5] The μ and λ parameters are not the same as those used for linear elastic transformations. We use the same notation anyway for simplicity

[6] The Reynold's number is a measure of the 'velocity' or complexity of the fluid motion.

onto the boundary is important later in this work, the *sliding* boundary conditions are chosen. They are defined for the $[0,1]^3$ domain (the unit cube) using the *Dirichlet* boundary conditions:

$$\begin{aligned} u_1(0,x_2,x_3) &= u_2(x_1,0,x_3) = u_3(x_1,x_2,0) = 0 \\ u_1(1,x_2,x_3) &= u_2(x_1,1,x_3) = u_3(x_1,x_2,1) = 0 \end{aligned} \tag{4.87}$$

and *Neumann* boundary conditions:

$$\begin{aligned} \left.\frac{\partial u_1}{\partial x_2}\right|_{(x_1,0,x_3)} &= \left.\frac{\partial u_1}{\partial x_2}\right|_{(x_1,1,x_3)} = 0 \\ \left.\frac{\partial u_1}{\partial x_3}\right|_{(x_1,x_2,0)} &= \left.\frac{\partial u_1}{\partial x_3}\right|_{(x_1,x_2,1)} = 0 \\ \left.\frac{\partial u_2}{\partial x_1}\right|_{(0,x_2,x_3)} &= \left.\frac{\partial u_2}{\partial x_1}\right|_{(1,x_2,x_3)} = 0 \\ \left.\frac{\partial u_2}{\partial x_3}\right|_{(x_1,x_2,0)} &= \left.\frac{\partial u_2}{\partial x_3}\right|_{(x_1,x_2,1)} = 0 \\ \left.\frac{\partial u_3}{\partial x_2}\right|_{(0,x_2,x_3)} &= \left.\frac{\partial u_3}{\partial x_2}\right|_{(1,x_2,x_3)} = 0 \\ \left.\frac{\partial u_3}{\partial x_3}\right|_{(x_1,0,x_3)} &= \left.\frac{\partial u_3}{\partial x_3}\right|_{(x_1,1,x_3)} = 0 \end{aligned} \tag{4.88}$$

The sliding boundary conditions map the unit cube onto the unit cube, in such a way that boundary points are allowed to slide along the boundary.

With these boundary conditions the eigen-functions and eigen-values have been derived by Miller, Christensen, et al. [Mil93, Chr94, Chr94c]. See [Chr94c] for the complete derivation. To simplify the equations a set of help functions are first defined:

$$\begin{aligned} \mathrm{scc}(\boldsymbol{x}) &= \sin(i\pi x_1)\cos(j\pi x_2)\cos(k\pi x_3) \\ \mathrm{csc}(\boldsymbol{x}) &= \cos(i\pi x_1)\sin(j\pi x_2)\cos(k\pi x_3) \\ \mathrm{ccs}(\boldsymbol{x}) &= \cos(i\pi x_1)\cos(j\pi x_2)\sin(k\pi x_3) \end{aligned} \tag{4.89}$$

With these definitions the orthogonal eigen-functions are given as:

$$\begin{aligned} \boldsymbol{\phi}_{ijk1}(\boldsymbol{x}) &= \alpha_1 \begin{bmatrix} i\,\mathrm{scc}(\boldsymbol{x}) \\ j\,\mathrm{csc}(\boldsymbol{x}) \\ k\,\mathrm{ccs}(\boldsymbol{x}) \end{bmatrix} \\ \boldsymbol{\phi}_{ijk2}(\boldsymbol{x}) &= \alpha_2 \begin{bmatrix} -j\,\mathrm{scc}(\boldsymbol{x}) \\ i\,\mathrm{csc}(\boldsymbol{x}) \\ 0 \end{bmatrix} \\ \boldsymbol{\phi}_{ijk3}(\boldsymbol{x}) &= \alpha_3 \begin{bmatrix} ik\,\mathrm{scc}(\boldsymbol{x}) \\ jk\,\mathrm{csc}(\boldsymbol{x}) \\ -(i^2+j^2)\,\mathrm{ccs}(\boldsymbol{x}) \end{bmatrix} \end{aligned} \tag{4.90}$$

The eigen-values corresponding to the eigen-functions are:

$$\begin{aligned} \kappa_{ijk1} &= -\pi^2(2\mu+\lambda)(i^2+j^2+k^2) \\ \kappa_{ijk2} &= \kappa_{ijk3} = -\pi^2\mu(i^2+j^2+k^2) \end{aligned} \tag{4.91}$$

For $i = j = k = 0$ the α_i are zero, otherwise they are chosen to ensure that the eigen-functions are normalized, ie. the inner product of each eigen-function with itself is one:

$$< \boldsymbol{\phi}_{ijkm}, \boldsymbol{\phi}_{ijkm} > = \int_\Omega \boldsymbol{\phi}^T_{ijkm} \boldsymbol{\phi}_{ijkm} d\boldsymbol{x} \; = \; 1 \tag{4.92}$$

For $i + j + k > 0$ this is obtained when:

$$\begin{aligned} \alpha_1 &= \sqrt{\frac{8}{\Gamma_{ijk}(i^2+j^2+k^2)}} \\ \alpha_2 &= \sqrt{\frac{8}{\Gamma_{ijk}(i^2+j^2)}} \\ \alpha_3 &= \sqrt{\frac{8}{\Gamma_{ijk}(i^2+j^2)(i^2+j^2+k^2)}} \end{aligned} \tag{4.93}$$

where

$$\Gamma_{ijk} = \begin{cases} 1 & \text{if none of } i,j,k \text{ are zero} \\ 2 & \text{if one of } i,j,k \text{ are zero} \\ 4 & \text{if two of } i,j,k \text{ are zero} \end{cases} \tag{4.94}$$

The function Γ was not included in the work of Miller, Christensen, et al. [Chr94c], but in the non-general cases, where one or more of the i, j, k parameters are zero, it cannot be omitted.

Using the orthonormal eigen-function basis we write the displacement field as:

$$\boldsymbol{u}(\boldsymbol{x}) = \lim_{IJK\to\infty} \sum_{i=0}^{I}\sum_{j=0}^{J}\sum_{k=0}^{K}\sum_{r=1}^{3} \nu_{ijkr}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) \tag{4.95}$$

where ν_{ijkr} are the eigen-function coefficients for the displacement field.

The number of basis functions in the decomposition of the displacement field is determined by I, J and K. The finite truncation of the series to the common N is denoted by:

$$\boldsymbol{u}_N(\boldsymbol{x}) = \sum_{i,j,k=0}^{N}\sum_{r=1}^{3} \nu_{ijkr}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) \tag{4.96}$$

We now apply the linear operator $\mathcal{L}$ to the displacement field:

$$\begin{aligned} \mathcal{L}\boldsymbol{u}_N(\boldsymbol{x}) &= \mathcal{L}\sum_{ijkr} \nu_{ijkr}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) = \sum_{ijkr} \nu_{ijkr}\mathcal{L}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) \\ &= \sum_{ijkr} \nu_{ijkr}\kappa_{ijkr}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) \end{aligned} \tag{4.97}$$

where we used the linearity of the operator.

4.6.1 Projection of forces onto the eigen-function basis

Let us consider the static elastic deformation equation using linear elasticity:

$$\mathcal{L}\boldsymbol{u}(\boldsymbol{x}) + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{o} \tag{4.98}$$

where $\boldsymbol{f}$ is the applied force field. Using the truncated decomposition in equation 4.97 we rewrite this as:

$$\sum_{ijkr} \nu_{ijkr}\kappa_{ijkr}\boldsymbol{\phi}_{ijkr}(\boldsymbol{x}) + \boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{o} \tag{4.99}$$

We take the inner product $< \boldsymbol{a}, \boldsymbol{b} >= \int \boldsymbol{a}^T \boldsymbol{b} d\boldsymbol{x}$ of the equation with $\boldsymbol{\phi}_{lmns}(\boldsymbol{x})$ and get:

$$\begin{aligned}
&\sum_{ijkr} \nu_{ijkr}\kappa_{ijkr} < \boldsymbol{\phi}_{ijkr}(\boldsymbol{x}), \boldsymbol{\phi}_{lmns}(\boldsymbol{x}) > + < \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\phi}_{lmns}(\boldsymbol{x}) > = \boldsymbol{o} \\
&\Updownarrow \\
&\nu_{lmns}\kappa_{lmns} + < \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\phi}_{lmns}(\boldsymbol{x}) > = \boldsymbol{o} \\
&\Updownarrow \\
&\nu_{lmns} = -\frac{1}{\kappa_{lmns}} < \boldsymbol{f}(\boldsymbol{x}), \boldsymbol{\phi}_{lmns}(\boldsymbol{x}) >
\end{aligned} \tag{4.100}$$

where the fact, that $< \boldsymbol{\phi}_{ijkr}(\boldsymbol{x}), \boldsymbol{\phi}_{lmns}(\boldsymbol{x}) >$ is zero for $(ijkr) \neq (lmns)$, was used to pick out the $lmns$ element of the summation, in the step from line 1 to line 2. We call equation 4.100 the *decomposition projection equation.*

This result is interesting, since it provides us with a simple way of determining the coefficients for the eigen-function decomposition of the displacement field given a force $\boldsymbol{f}$. Indeed the eigen-basis coefficients are simply the projection by the inner product of the force onto the eigen-functions, scaled by the inverse of the eigen-values.

4.6.2 2D eigen-function basis

For completeness we also give the eigen-functions and eigen-values for the 2D version of the linear elastic operator:

$$\mathcal{L}\boldsymbol{u} = \mu\boldsymbol{\Delta}\boldsymbol{u}(\boldsymbol{x}) + (\mu + \lambda)\boldsymbol{\nabla}(\boldsymbol{\nabla} \cdot \boldsymbol{u}(\boldsymbol{x})) \tag{4.101}$$

With the sliding boundary conditions on the domain $[0, 1]^2$:

$$u_1(0, x_2) = u_2(x_1, 0) = 0 \tag{4.102}$$

$$\begin{aligned}
u_1(1, x_2) &= u_2(x_1, 1) = 0 \\
\left.\frac{\partial u_1}{\partial x_2}\right|_{(x_1,0)} &= \left.\frac{\partial u_1}{\partial x_2}\right|_{(x_1,1)} = 0 \\
\left.\frac{\partial u_2}{\partial x_1}\right|_{(0,x_2)} &= \left.\frac{\partial u_2}{\partial x_1}\right|_{(1,x_2)} = 0
\end{aligned} \tag{4.103}$$

the eigen-functions are:

$$\begin{aligned} \boldsymbol{\phi}_{ij1}(\boldsymbol{x}) &= \alpha_1 \begin{bmatrix} i\,\mathrm{sc}(\boldsymbol{x}) \\ j\,\mathrm{cs}(\boldsymbol{x}) \end{bmatrix} \\ \boldsymbol{\phi}_{ij2}(\boldsymbol{x}) &= \alpha_2 \begin{bmatrix} -j\,\mathrm{sc}(\boldsymbol{x}) \\ i\,\mathrm{cs}(\boldsymbol{x}) \end{bmatrix} \end{aligned} \tag{4.104}$$

where

$$\begin{aligned} \mathrm{sc}(\boldsymbol{x}) &= \sin(i\pi x_1)\cos(j\pi x_2) \\ \mathrm{cs}(\boldsymbol{x}) &= \cos(i\pi x_1)\sin(j\pi x_2) \end{aligned} \tag{4.105}$$

The eigen-values corresponding to the eigen-functions are:

$$\begin{aligned} \kappa_{ij1} &= -\pi^2(2\mu+\lambda)(i^2+j^2) \\ \kappa_{ij2} &= -\pi^2\mu(i^2+j^2) \end{aligned} \tag{4.106}$$

For $i = j = 0$ the α_i are all zero, otherwise they are given by:

$$\alpha_1 = \alpha_2 = \sqrt{\frac{4}{\Gamma_{ij}(i^2+j^2)}} \tag{4.107}$$

$$\Gamma_{ij} = \frac{1}{2}\Gamma_{ij0} \tag{4.108}$$

4.7 Summary

This chapter has presented the joint theoretical basis of elastic and viscous fluid models. A comprehensive understanding of this theory is necessary to take full advantage of the continuum models and to avoid the pitfalls created by the many assumptions used in the field.

The following chapters all build on this theory, and thus refer to this chapter repeatedly.

Chapter 5

Non-rigid Registration using Continuum Models

The subject of this chapter is non-rigid registration using physical continuum models. The first section defines the non-rigid registration problem. In the second section elastic registration methods are discussed.

Some of these methods use linear models of elastic deformation. Unfortunately, registration using linear elasticity often violates the inherent small deformation assumption. For large deformations the topology cannot be guaranteed and an alternative approach is needed.

The next section describes such an alternative registration method using viscous fluid continuum models, which were originally proposed by Christensen et al. [Chr93, Chr94b, Chr94c, Chr96]. These models have the advantage that they allow large complex deformations and implement free registration as opposed to elastic registration.

The chapter then proceeds to propose a new algorithm for viscous fluid registration, which is at least an order of magnitude faster than the previous algorithm by Christensen et al. This speedup is important, since it allows the viscous fluid registration algorithm to be run on an ordinary single-processor workstation, instead of the massively parallel computer needed by the algorithm of Christensen et al.

The algorithm uses convolution with a new filter which implements the linear elasticity operator described in the previous chapter.

Thirion [Thi96] has recently proposed a 'demon'-based algorithm for free registration of medical images. We show that the work of Thirion is actually a primitive version of the fluid registration algorithm, and discuss the possible consequences of the use of the Gaussian filter in Thirion's work, instead of our linear elasticity filter. The advantage of the Gaussian filter is that it is separable and thus allows fast computation.

5.1 Defining the problem

Following the definitions of Christensen et al. [Chr93, Chr94b, Chr94c, Chr96] the non-rigid registration problem is regarded as a question of mapping or warping a template image onto an individual study image.

The template image can be part of an anatomical atlas[1], which possibly in addition to the template image includes other modalities, topological information, functional information, etc.

Denote the template image $T(\boldsymbol{x})$ and the study image $S(\boldsymbol{x})$ where $\boldsymbol{x} \in [0,1]^3$. Formulated in the Lagrangian reference frame the purpose of the registration is consequently to find a transformation $\boldsymbol{\varphi}(\boldsymbol{x}) = \boldsymbol{x} + \boldsymbol{u}(\boldsymbol{x})$ that maps $T(\boldsymbol{x})$ onto $S(\boldsymbol{x})$, in such a way that $T(\boldsymbol{x})$ matches $S(\boldsymbol{\varphi}(\boldsymbol{x}))$. In the Eulerian reference frame we seek a transformation $\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}) = \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}})$ so that $T(\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}))$ matches $S(\tilde{\boldsymbol{x}})$.

The transformation $\boldsymbol{\varphi}(\boldsymbol{x}) : \Omega \rightarrow \tilde{\Omega}$ must be *homeomorphic* to ensure that the transformation maintains the topology and structure of the original template image. A homeomorphic mapping is defined as being *continuous*, *1-to-1* and *onto*. A thorough discussion of the homeomorphism can be found in Christensen [Chr94c].

In practice, to ensure that the transformation is homeomorphic, we demand that the boundary of the domain is mapped onto itself, and that the Jacobian of the transformation is greater than zero everywhere. The boundary requirement is satisfied, when fixed or sliding boundary conditions are employed, as they are with the linear elastic eigen-basis introduced in the section 4.6. Also note that the concatenation of homeomorphic mappings yields a homeomorphic mapping.

5.2 Elastic registration

Bajcsy et al. [Baj89, Gee93] were the first to describe 3D voxel-based non-rigid registration of medical images using elastic models. Building on initial work by Broit [Brt81], they modelled the template image as a linear elastic solid and deformed it using forces derived from an approximation of the local gradient of a correlation based similarity measure. In addition they used multi-resolution to increase the speed.

Miller, Christensen, et al. [Mil93, Chr94] also used the linear elastic deformation model, but applied a Gaussian sensor model, ie. a least-squares measure of the grey-level differences in the two images. To regularize the registration problem they first registered the images using a truncated linear elastic eigenbasis (as defined in section 4.6), and then followed this initial registration with a registration using the full linear elastic model.

Christensen et al. [Chr94c] later realized, that using the linear elastic continuum model was not appropriate, since the assumption of infinitisimal deformation was frequently violated for the kind of deformation that is required in non-rigid registration. The transformation is not guaranteed to be homeomorphic and, therefore, not well-behaved.

[1] Called anatomical textbook in Christensen et al. [Chr93, Chr94b, Chr94c, Chr96]

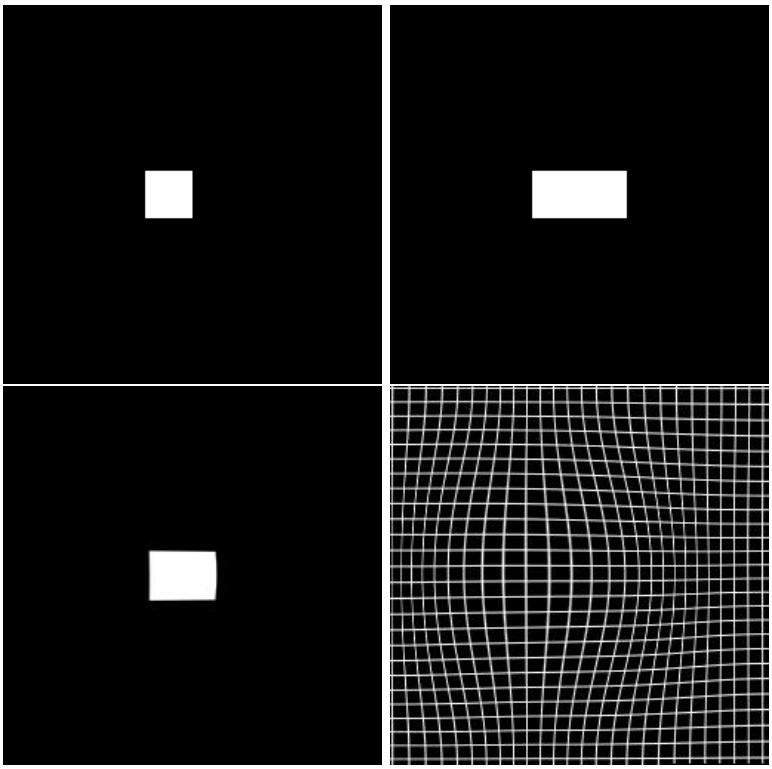

Figure 5.1: *The result of registering a 2D square to a rectangle using linear elastic registration. Top: Template and study image. Bottom: Result and deformed grid.*

The group has since then proposed two non-rigid registration methods which satisfy the large deformation requirement. First Christensen et al. [Chr93, Chr94b, Chr94c, Chr96] proposed the fluid registration method which we will discuss in the next section, and later Rabbitt et al. [Rab95] proposed the use of a non-linear (unspecified) hyperelastic model, instead of the linear elastic model used previously in elastic registration. Rabbitt optimized a 2D example using non-linear Finite Elements.

Following our definitions, the fluid registration approach by Christensen et al. allows free registration, and the approach of Rabbitt et al. elastic registration. Both these methods are therefore useful and well-behaved algorithms in their respective classes of registration methods.

Examples of linear elastic and viscous fluid registration using the algorithms of Miller, Christensen, et al. [Mil93, Chr93, Chr94, Chr94b, Chr94c, Chr96] are shown in figures 5.1 and 5.2, where a 2D square is registered to a rectangle. Note, how the elastic effect limits the possible deformation of the elastic registration, whereas the fluid algorithm allows full registration of the template to the study

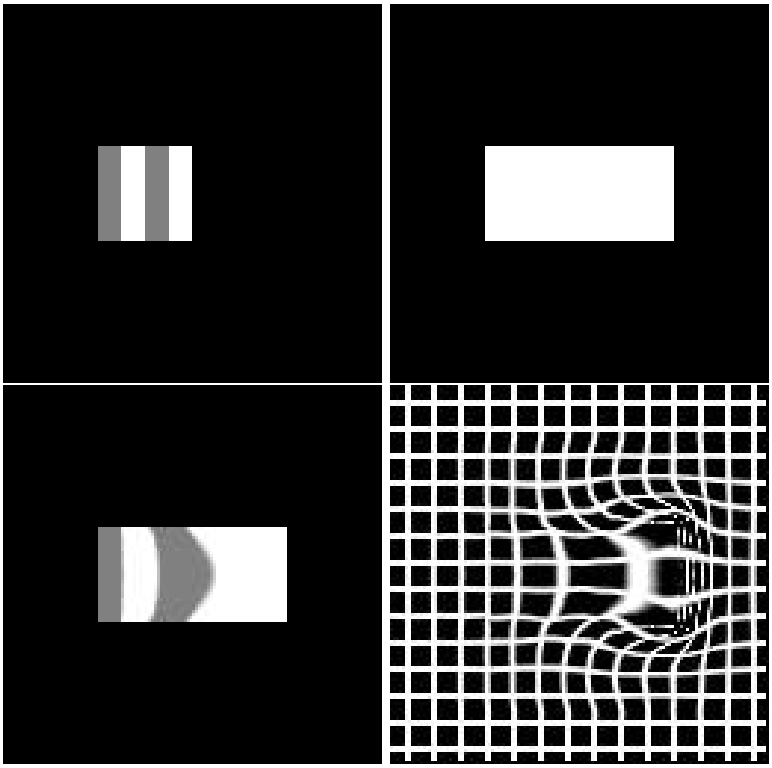

Figure 5.2: *The result of registering a 2D square to a rectangle using viscous fluid registration. Top: Template and study image. Bottom: Result and deformed grid.*

image. This was also illustrated in figure 2.1 in chapter 2.

The next section describes the viscous fluid registration algorithm used by Christensen et al. and then continues to propose a new and faster algorithm.

5.3 Viscous fluid registration

In [Chr93, Chr94b, Chr94c, Chr96] Christensen et al. described a free registration approach in which they use a viscous fluid model to control the deformation. The template image is modelled as a thick fluid which flows out to match the study, under the control of the same Gaussian sensor model used in [Chr94, Mil93]. In [Chr94c] Christensen argues that this Gaussian sensor model is theoretically better than correlation based similarity measures.

Unfortunately, the algorithm proposed by Christensen et al. is rather slow. They originally implemented the algorithm using a massively parallel DECmpp 128x64 MasPar computer, on which the algorithm used on the order of 5-10 minutes for 2D and 2-6 hours for 3D registrations. In a recent paper [Chr96] they show estimates of the execution time on a MIPS R4400 processor on the order of 2 hours for 2D and 7 days for 3D. In practice this means that the algorithm is not useful unless a massively parallel computer is available.

The contribution of this chapter is a new fast algorithm based entirely on convolution with filters, which gives a speedup of at least an order of magnitude.

The original viscous fluid algorithm by Christensen et al. [Chr93, Chr94b, Chr94c, Chr96] is first described. The viscous fluid model is introduced along with the force field and the numerical solution method. We discuss the core part of Christensen's numerical solution, and introduce the general idea behind the convolution approach, which is proposed to increase the speed of the method. And finally, the new filter for the convolution approach is derived.

5.3.1 Viscous fluid model

Following Christensen et al., the viscous fluid PDE defined in equation 4.85 is used. Note, that the viscous fluid PDE was derived here without introducing the *mass source* term η that Christensen et al. use. Although, they emphasize the introduction of this mass source term they quickly proceed to remove it, using the assumption of low Reynold's number flow (which is also used here). In the simplified models which both Christensen et al. and this text use, this mass source term is consequently without influence (although theoretically relevant).

In the Eulerian framework we can write the partial differential equation (PDE) for the viscous fluid deformation of the template as (equation 4.85):

$$\mu\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}) + (\mu+\lambda)\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}})) + \tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}})) = 0 \tag{5.1}$$

where $\tilde{\boldsymbol{\Delta}} = \tilde{\boldsymbol{\nabla}}^T\tilde{\boldsymbol{\nabla}}$ is the Laplacian operator, and μ and λ are controlling parameters. The force field $\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}))$ is used to drive the flow.

For a constant force $\tilde{\boldsymbol{f}}$ this is the PDE for linear elasticity working on the instantaneous velocity field $\tilde{\boldsymbol{v}}$. The equation, consequently, works by elasticly smoothing the instantaneous velocity field of the fluid in the Eulerian reference frame.

The term $\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}}$ is called the viscous term because it constrains the velocity field spatially. The $\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}})$ term allows for contraction or expansion of the fluid.

5.3.2 Force field

The force field is the link between the physical model of the fluid and the image data. It is crucial for the success of the registration algorithm, and can take on many forms as shown in chapter 3 on multimodality voxel similarity measures. In a statistical framework it is called the *sensor model*.

Bajcsy et al. [Baj89] used the local gradient of the normalized cross-correlation of a small region in the template image T with a neighbourhood in the study image S. To smooth the local cost-function they used a quadratic Taylor series expansion of the cost-function around each voxel. In addition the cross-correlation function were projected onto a truncated series of Hermite polynomials to enhance the response of particular features such as edges.

Miller, Christensen et al. [Mil93, Chr93, Chr94, Chr94b, Chr94c, Rab95, Chr96] have used a Gaussian sensor model throughout their work. For MRI images this Gaussian sensor model appears to be an appropriate model of the variation between the registered template and study images [Chr94c, McV85]. But for other modalities such as CT it might not be appropriate. It is used here for CT anyway because of it's simplicity.

The cost function for the Gaussian sensor model is written as the sum of squared differences between the two images:

$$C(T(\tilde{\boldsymbol{x}}), S(\tilde{\boldsymbol{x}}), \tilde{\boldsymbol{u}}) = \frac{1}{2} \int_{\tilde{\Omega}} |T(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)) - S(\tilde{\boldsymbol{x}})|^2 d\tilde{\boldsymbol{x}} \tag{5.2}$$

The expectation of this function is zero when the images are perfectly registered.

The force field is determined as the variation of the cost-function wrt. the displacement field:

$$\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)) = -(T(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)) - S(\tilde{\boldsymbol{x}}))\boldsymbol{\nabla} T|_{\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)} \tag{5.3}$$

This force function has two terms. The $\boldsymbol{\nabla} T|_{\boldsymbol{x} = \tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)}$ term is the gradient of the template image in the Lagrangian reference frame. It responds to intensity changes in the template image, such as edges and ridges, and can be compared to the *demons* in Thirion [Thi96], which act as 'traffic wardens' on the edge of image objects by determining the local force field.

The second term $T(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t)) - S(\tilde{\boldsymbol{x}})$ scales the first term, such that in positions with large intensity differences, movement is encouraged and in similar regions movement is discouraged. The force field is consequently zero in regions that are locally similar.

Although the Gaussian cost function is minimized, when the images are registered, the force field will not necessarily lead us to this minimum. There might be many local minima. In addition, since the force field for the Gaussian sensor model is derived based on only one voxel and the finite difference estimators footprint, there is likely to be more local minima than when the force field is determined using a small region as Bajcsy et al. do.

Region-based approaches like that of Bajcsy et al., which can be compared to the image profiles used by Cootes et al. [Coo95] in their Active Shape Models, use more information to build the local force field and can, consequently, be expected to be better and more robust.

On the other hand, region-based local force fields incur an added computational cost. Since the viscous fluid registration algorithm discussed here is already time consuming, we are forced to use the Gaussian sensor model.

5.3.3 Numerical solution

Solution of the viscous fluid registration problem requires solving the joint set of equations determined by the viscous fluid PDE, the material derivative (eq. 4.19), and the force field equation defined above [Chr94c]:

$$\mu\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t)+(\mu+\lambda)\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}}\cdot\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t))+\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t))=\boldsymbol{o} \tag{5.4}$$

$$\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}},\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t))=-(T(\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t))-S(\tilde{\boldsymbol{x}}))\boldsymbol{\nabla}T|_{\tilde{\boldsymbol{x}}-\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t)} \tag{5.5}$$

$$\frac{\partial\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t)}{\partial t}=\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t)-\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t)\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t) \tag{5.6}$$

This set of equations includes non-linearities in both the force and the material derivative. To solve it, Euler integration is applied over time, using a forward finite difference estimate of the time derivative in equation 5.6:

$$\begin{aligned}\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t_{i+1}) &= \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t_i)+(t_{i+1}-t_i)(\boldsymbol{I}-\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t_i))\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t_i)\\ &= \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}},t_i)+(t_{i+1}-t_i)\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}},t_i)\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}},t_i)\end{aligned} \tag{5.7}$$

The solution can be found by iteratively solving equation 5.4 for the instantaneous force determined by equation 5.5, and integrating over time using equation 5.7.

Regridding

In section 5.1 we discussed the requirements for the transformation to be well-formed, ie. homeomorphic. One of the requirements were that the transformation should have a non-zero Jacobian. From a theoretical point of view the Jacobian could be negative or positive, but to follow the definitions for the continuum models, we require the deformation to have a positive Jacobian, ie. $J=|\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}},t_i)|>0$.

The requirement for a non-zero Jacobian can also be seen from the Euler integration equation 5.7. Reliable Euler integration requires a well-conditioned

transformation gradient $\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t_i)$. Since the Jacobian provides a measure of the condition of $\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t_i)$ we get the same requirement that $J \neq 0$.

The transformation becomes singular for large curved transformations. To evade this problem, the regridding method of Christensen [Chr94c] is used. Every time the Jacobian J drops below 0.5 a new template is generated by applying the current deformation. The displacement field is set to zero, whereas the current velocities remain unchanged. The total deformation becomes the concatenation of the displacement fields, associated with the sequence of propagated templates. Remember, that the concatenation of homeomorphic mappings is a homeomorphic mapping.

Algorithm

The complete algorithm for solving the viscous fluid registration problem becomes [Chr94c]:

1. Let $i = 0$ and $\tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, 0) = 0$
2. Calculate the body force $\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t_i))$ using equation 5.5.
3. If $\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t_i))$ is below a threshold for all $\tilde{\boldsymbol{x}}$, then STOP.
4. Solve the linear PDE equation 5.4 for instantaneous velocity $\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}, t_i)$ and force $\tilde{\boldsymbol{f}}(\tilde{\boldsymbol{x}}, \tilde{\boldsymbol{u}}(\tilde{\boldsymbol{x}}, t_i))$.
5. Choose a timestep $(t_{i+1} - t_i)$ so that $\|(t_{i+1} - t_i)\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t_i)\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}, t_i)\| < d\tilde{\boldsymbol{u}}_{max}$, where $d\tilde{\boldsymbol{u}}_{max}$ is the maximal flow distance allowed in one iteration (0.7 in this work).
6. Prepare for Euler integration using equation 5.7.
7. If the Jacobian $J = |\tilde{\boldsymbol{\nabla}}\tilde{\boldsymbol{\varphi}}(\tilde{\boldsymbol{x}}, t_i)|$ becomes less than 0.5 with the prepared Euler integration, then regrid the template.
8. Execute the Euler integration using equation 5.7.
9. $i = i + 1$, goto 2

The only remaining question is how to solve the PDE equation in step 4. We discuss this in the following section.

5.3.4 Solving the linear PDE

In the algorithm shown above, the core problem is solving the linear PDE:

$$\tilde{\mathcal{L}}\tilde{\boldsymbol{v}} = \mu\tilde{\boldsymbol{\Delta}}\tilde{\boldsymbol{v}} + (\mu + \lambda)\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}}) = -\tilde{\boldsymbol{f}} \tag{5.8}$$

for constant force and time. In practice, solving this PDE is the most time consuming part of the fluid registration. The contribution of this chapter is a fast way of doing this.

As seen previously, for constant force $\tilde{\boldsymbol{f}}$ the PDE is linear, and the linear operator $\tilde{\mathcal{L}}$ is the linear elasticity operator working on $\tilde{\boldsymbol{v}}$. Linear elastic problems are normally solved using implicit finite element or finite difference methods. But in the case of images, we assign nodes in the elastic model to each pixel or voxel. The size of the problem is, therefore, huge and in practice unsolvable with these techniques.

Christensen et al. [Chr93, Chr94b, Chr94c, Chr96] used successive overrelaxation (SOR), with checker board update, to solve the linear elastic problem.

We suggest solving the linear PDE with multi-resolution convolution. Using the linearity of the PDE and the superposition principle, we create a filter as the impulse response of the linear operator $\tilde{\mathcal{L}}$ and, subsequently, apply this filter to the force field $\tilde{\boldsymbol{f}}$.

There is an algorithmic simularity between the SOR algorithm and the convolution approach proposed here, since the SOR algorithm can be seen as recursive convolution with a small filter. But, where the SOR algorithm requires many iterations to find the solution of the PDE, our approach gives a closed-form solution directly.

This work has been inspired by the work of Nielsen et al. [Nie94], who show that Tikhonov regularization can be implemented using Gaussian scale-space, and Thirion [Thi96], who propose a 'demon'-based registration algorithm, which we will demonstrate later, is an approximation to the viscous fluid registration problem.

5.3.5 Convolution filter for linear elasticity

In this section, we develop the convolution filter used to solve the linear PDE. First the impulse response of the linear operator is determined in the linear elasticity eigen-basis. The impulse response of a linear operator is a filter that implements the operator. Finally, the impulse response is discretized to get a discrete filter.

Determining the impulse response of $\tilde{\mathcal{L}}$

The impulse response in the direction of $\tilde{x}_1$ is determined as the displacement field corresponding to an impulse force $\tilde{\boldsymbol{f}}_c$ applied in the middle $\tilde{\boldsymbol{x}}_c$ of the domain:

$$\tilde{\boldsymbol{x}}_c = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \tag{5.9}$$

$$\tilde{\boldsymbol{f}}_c = \begin{bmatrix} \delta(\tilde{\boldsymbol{x}} - \tilde{\boldsymbol{x}}_c) \\ 0 \\ 0 \end{bmatrix} \tag{5.10}$$

$$\tag{5.11}$$

where $\delta(\tilde{\boldsymbol{x}})$ is Dirac's delta.

Using the result of section 4.6.1 in the previous chapter, we apply the decomposition projection equation 4.100 to the impulse force $\tilde{\boldsymbol{f}}_c$ to get the decomposition coefficients of the impulse response:

$$\begin{aligned} \nu_{lmns} &= -\frac{1}{\kappa_{lmns}} < \tilde{\boldsymbol{f}}_c, \boldsymbol{\phi}_{lmns}(\tilde{\boldsymbol{x}}) > \\ &= -\frac{1}{\kappa_{lmns}} \phi^{\tilde{x}_1}_{lmns}(\tilde{\boldsymbol{x}}_c) \end{aligned} \tag{5.12}$$

where $\phi^{\tilde{x}_1}_{lmns}$ is the $\tilde{x}_1$-coordinate of $\boldsymbol{\phi}_{lmns}$.

The coefficients are inserted into the equation for the decomposition of the displacement field. After some rearrangement of the terms, which is shown in appendix B, the result becomes:

$$\begin{aligned} \tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}) &= \sum_{ijk}^{\infty} \sum_{r=1}^{3} \nu_{ijkr} \boldsymbol{\phi}_{ijkr}(\tilde{\boldsymbol{x}}) \\ &= \sum_{ijk}^{\infty} \sum_{r=1}^{3} -\frac{1}{\kappa_{ijkr}} \phi^{\tilde{x}_1}_{ijkr}(\tilde{\boldsymbol{x}}_c) \boldsymbol{\phi}_{ijkr}(\tilde{\boldsymbol{x}}) \\ &= \frac{8}{\pi^2 \mu(2\mu+\lambda)} \sum_{ijk=0}^{\infty} \frac{\mathrm{scc}(\tilde{\boldsymbol{x}}_c)}{(i^2+j^2+k^2)^2 \Gamma_{ijk}} \\ &\quad \begin{bmatrix} (\mu i^2 + (2\mu+\lambda)(j^2+k^2))\mathrm{scc}(\tilde{\boldsymbol{x}}) \\ -ij(\mu+\lambda)\mathrm{csc}(\tilde{\boldsymbol{x}}) \\ -ik(\mu+\lambda)\mathrm{ccs}(\tilde{\boldsymbol{x}}) \end{bmatrix} \end{aligned} \tag{5.13}$$

This equation gives us the response of the linear operator $\tilde{\mathcal{L}}$ for an impulse force in the $\tilde{x}_1$ direction applied in $\tilde{\boldsymbol{x}}_c$, ie. the impulse response. The impulse response for the other directions are determined by simple rotation of the response for the $\tilde{x}_1$ direction. In the next section, we will see how this impulse response can be used to determine a discrete filter implementing the linear operator $\tilde{\mathcal{L}}$.

Discretizing the impulse response

In general the impulse response of the linear operator is the linear filter implementing the operator. In the continuous case, a force applied to a single point yields an infinitely large displacement of this particular point. However, in the discrete case we sample the filter on a discrete grid, and apply a lowpass filtering with a cut-off at the Nyquist frequency, to eliminate aliasing from higher order frequency components. The force is thereby smoothed over a small area or volume.

We note that the decomposition of the impulse response based on the eigenfunction basis is a frequency based decomposition. Large i, j and k correspond to high frequencies and small to low frequencies. An ideal lowpass filtering of the impulse response can, therefore, be performed by truncating the sequence at N instead of summing to infinity.

The sampled filter is defined with dimensions $D \times D \times D$, D odd, in the domain $[0,1]^3$. The sampling interval is consequently $\theta = 1/(D-1)$ which Shannon's sampling theorem relates to the cut-off frequency f by $\theta \leq 1/2f$. From equation 5.13 the frequencies corresponding to the summation variables are determined:

$$f_i = \frac{1}{2}i \qquad f_j = \frac{1}{2}j \qquad f_k = \frac{1}{2}k \tag{5.14}$$

and the common truncation point becomes $i = j = k = N = D-1$. We gather everything in:

Theorem 5.1 *Consider a 3D filter of size $D \times D \times D$, D odd, and let the lattice be addressed by $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \tilde{y}_2, \tilde{y}_3]^T$, where $\tilde{y}_i \in [-\frac{D-1}{2}, \frac{D-1}{2}] \cap \mathbb{N}$, $r = 1,2,3$. The filter implementing the 3D linear elastic operator $\tilde{\mathcal{L}}$ for the $\tilde{x}_1$ direction is then:*

$$\begin{aligned}\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}) \quad = \quad & \frac{8}{\pi^2\mu(2\mu+\lambda)} \sum_{ijk=0}^{D-1} \frac{scc(\tilde{\boldsymbol{x}}_c)}{(i^2+j^2+k^2)^2\Gamma_{ijk}} \\ & \begin{bmatrix} (\mu i^2 + (2\mu+\lambda)(j^2+k^2))scc(\tilde{\boldsymbol{x}}) \\ -ij(\mu+\lambda)csc(\tilde{\boldsymbol{x}}) \\ -ik(\mu+\lambda)ccs(\tilde{\boldsymbol{x}}) \end{bmatrix}\end{aligned}$$

where $scc(\tilde{\boldsymbol{x}})$, $csc(\tilde{\boldsymbol{x}})$, and $ccs(\tilde{\boldsymbol{x}})$ are defined in section 4.6, and

$$\tilde{\boldsymbol{x}} = \frac{1}{D-1}\tilde{\boldsymbol{y}} + \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \quad \tilde{\boldsymbol{x}}_c = \begin{bmatrix} 0.5 \\ 0.5 \\ 0.5 \end{bmatrix} \tag{5.15}$$

□

A filter for 2D linear elasticity can be derived in a similar fashion (see Bro-Nielsen and Gramkow [BN96d]):

Theorem 5.2 *Consider a 2D filter of size $D \times D$, D odd and let the lattice be addressed by $\tilde{\boldsymbol{y}} = [\tilde{y}_1, \tilde{y}_2]^T$, where $\tilde{y}_i \in [-\frac{D-1}{2}, \frac{D-1}{2}] \cap \mathbb{N}$, $r = 1,2$. The filter implementing the 2D linear elastic operator $\tilde{\mathcal{L}}$ for the $\tilde{x}_1$ direction is then:*

$$\begin{aligned}\tilde{\boldsymbol{v}}(\boldsymbol{x}) = \frac{4}{\pi^2\mu(2\mu+\lambda)} \quad & \sum_{ij=0}^{D-1} \quad \frac{sc(\tilde{\boldsymbol{x}}_c)}{(i^2+j^2)^2\Gamma_{ij}} \\ & \begin{bmatrix} -j^2(i^2\mu + (2\mu+\lambda))sc(\tilde{\boldsymbol{x}}) \\ ij(\mu+\lambda)cs(\tilde{\boldsymbol{x}}) \end{bmatrix}\end{aligned} \tag{5.16}$$

where $sc(\tilde{\boldsymbol{x}})$ and $cs(\tilde{\boldsymbol{x}})$ are defined in section 4.6.2, and

$$\tilde{\boldsymbol{x}} = \frac{1}{D-1}\tilde{\boldsymbol{y}} + \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \quad \tilde{\boldsymbol{x}}_c = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix} \tag{5.17}$$

□

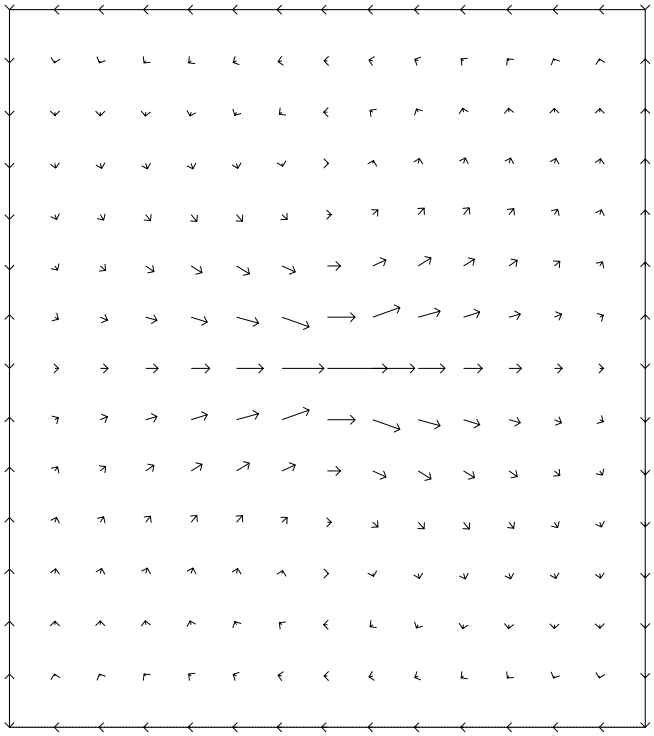

Figure 5.3: *Displacements of 2D linear elastic filter.*

We leave it to the reader to determine the filter components for the other directions.

A plot of the displacement field of the 2D filter is shown in figure 5.3. Note, how the filter pulls in material vertically behind the horizontal impulse force. This orthogonal effect is due to the $\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}))$ term in the linear elastic deformation equation and, is important for the viscous effect of the fluid.

To show that the application of the filter for linear elastic deformation works, we have made some experiments using the filter as the linear elasticity operator, and compared the results with a Finite Element Model (FEM) of linear elasticity. The results show quite similar deformations. A comparison for a deformation corresponding to forces applied to 6 nodes in the domain is shown in figure 5.4[2].

5.3.6 Multi-resolution implementation

Because of the limited span of the discrete filter, we have implemented the viscous fluid registration algorithm using the filter in multi-resolution. The fluid registration is first performed on a rough scale. The result of this scale is then propagated to a finer scale and the fluid registration restarted here. This process is continued down to the finest scale of the scale-space, yielding the final registration result. All the results shown in this chapter used 3 levels of resolution.

5.3.7 Results

In the previous sections the original theory of the viscous fluid registration method has been described, and a convolution filter was developed to replace the linear operator in the core routine of the fluid registration.

Figure 5.2 shows the result of registering a square to a rectangle. Figure 5.5 show the result of registering a circle to a 'C' using viscous fluid deformation. The grid shows the curved and very large deformations that are applied to the template. Although the deformation is very large, the topology of the template is maintained as shown by the rings in the 'bull's eye'. Figure 5.6 shows the developing deformation as the fluid circle deforms into the 'C'. These results are very similar to figures 10.20-23 in [Chr94c].

Figure 5.7 shows the results for two adjacent CT slices. Topologically similar regions have been registered correctly. But, if the images are studied closely, they show that the topology of the original image has been retained to such an extent, that a full registration of the topologically different regions has not been achieved. This is a practical example of the topology-maintaining characteristics of the viscous fluid registration algorithm, ie. the homeomorphic mapping. A similar example of registration of two adjacent MR slices is shown in figure 5.8.

Figure 5.9 shows a practical application of 3D fluid registration. In this case 3 jaws, from the same patient at different times, have been extracted from

[2]The calculations are not exactly comparable since the finite element model used fixed displacements on the boundary instead of the sliding boundary conditions used by the linear elastic filter.

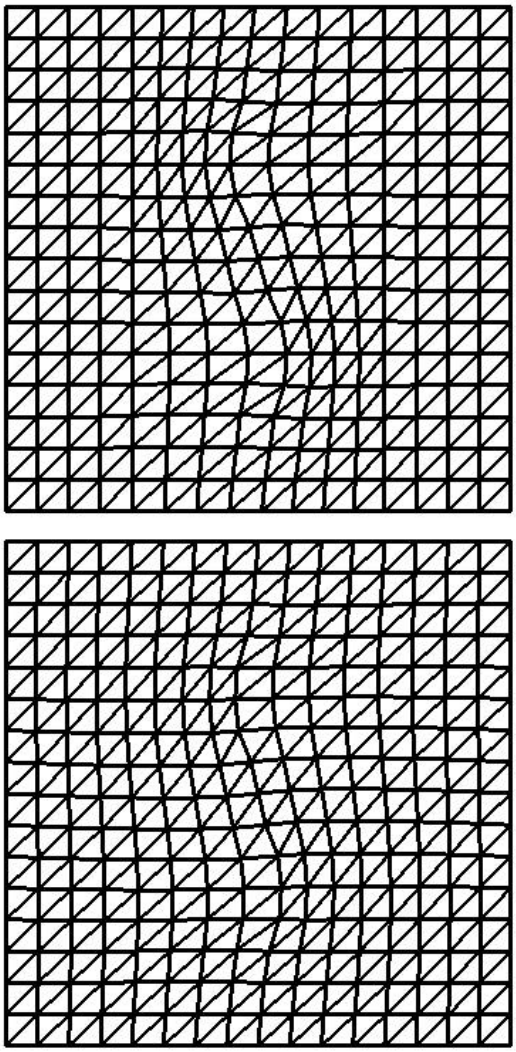

Figure 5.4: *Complex deformation resulting from three leftward forces in (0,2), (0,3) and (0,4), and three rightward forces in (0,-2), (0,-3) and (0,-4), the center is (0,0). Left: Using filter. Right: Using FEM.*

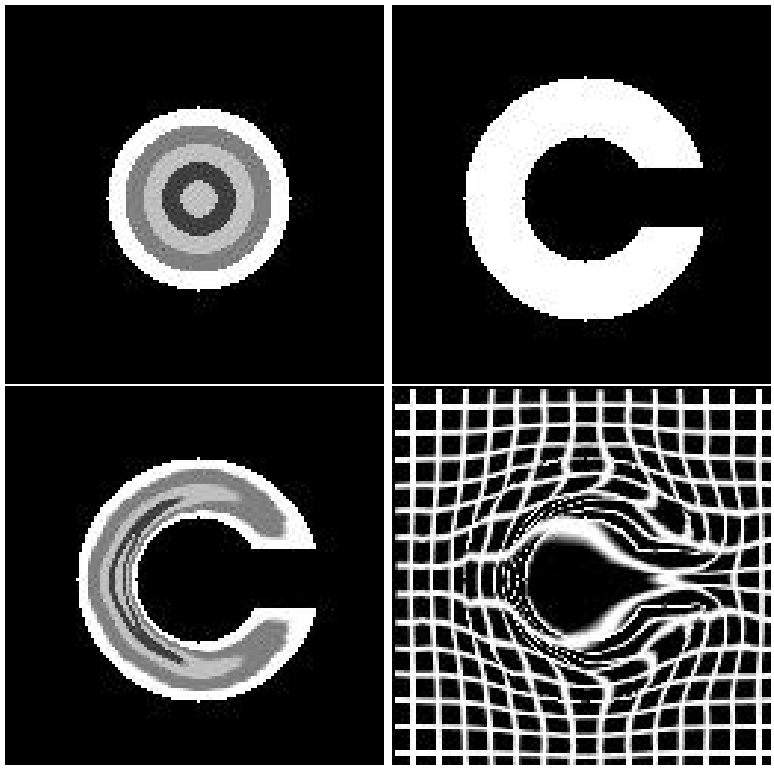

Figure 5.5: *Circle deforming to a 'C' using viscous fluid registration. Top: Template and study images. Bottom: Result and deformed grid.*

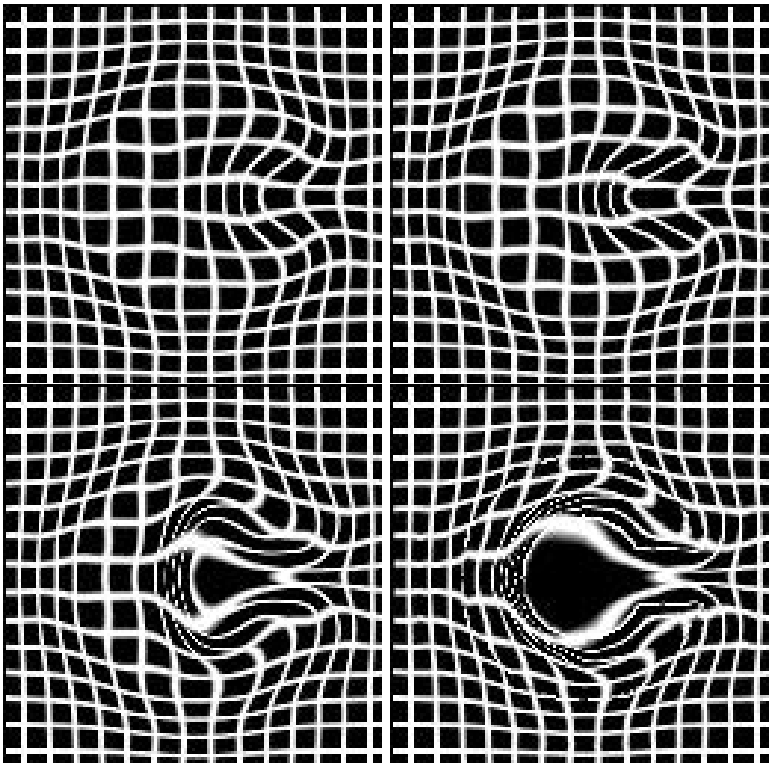

Figure 5.6: *Development of deformation of grid for viscous fluid registration of circle to 'C'.*

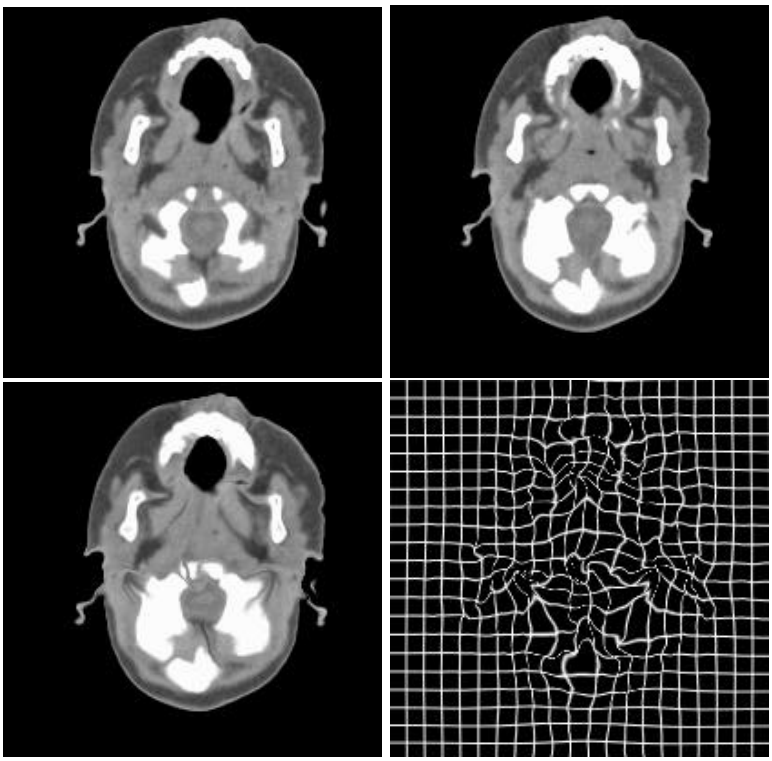

Figure 5.7: *CT slice registered to another slice using viscous fluid model. Top: Template and study images. Bottom: Result and deformed grid.*

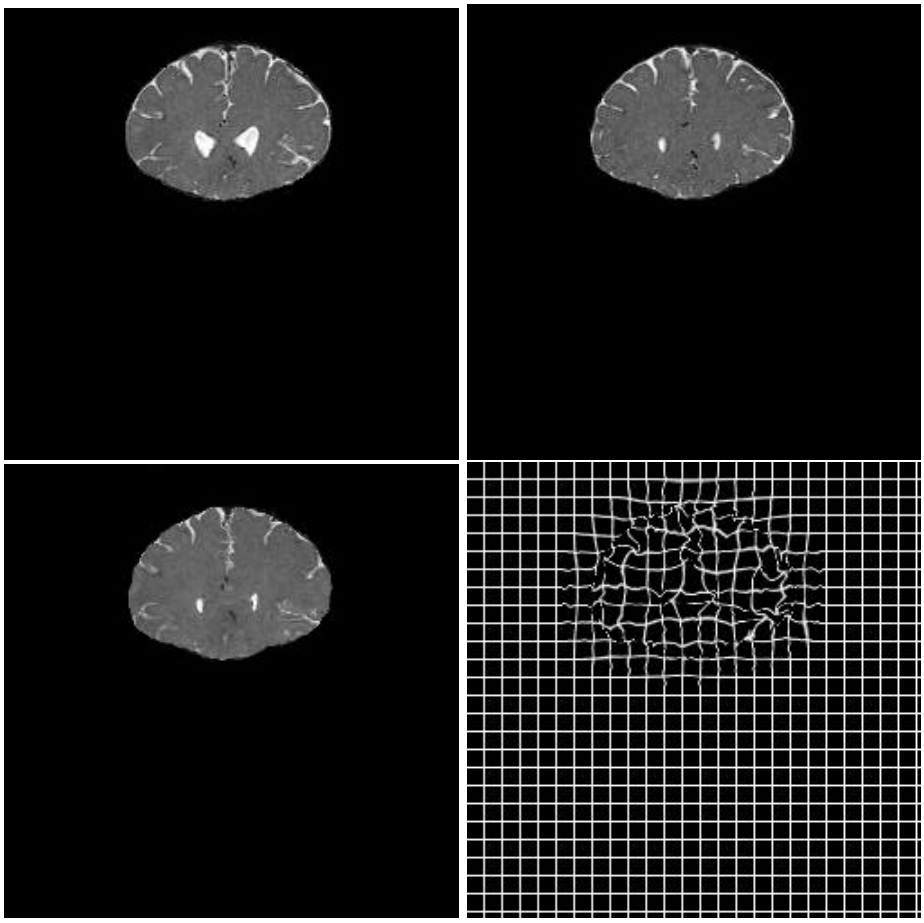

Figure 5.8: *MR slice registered to another slice using viscous fluid model. Top: Template and study images. Bottom: Result and deformed grid.*

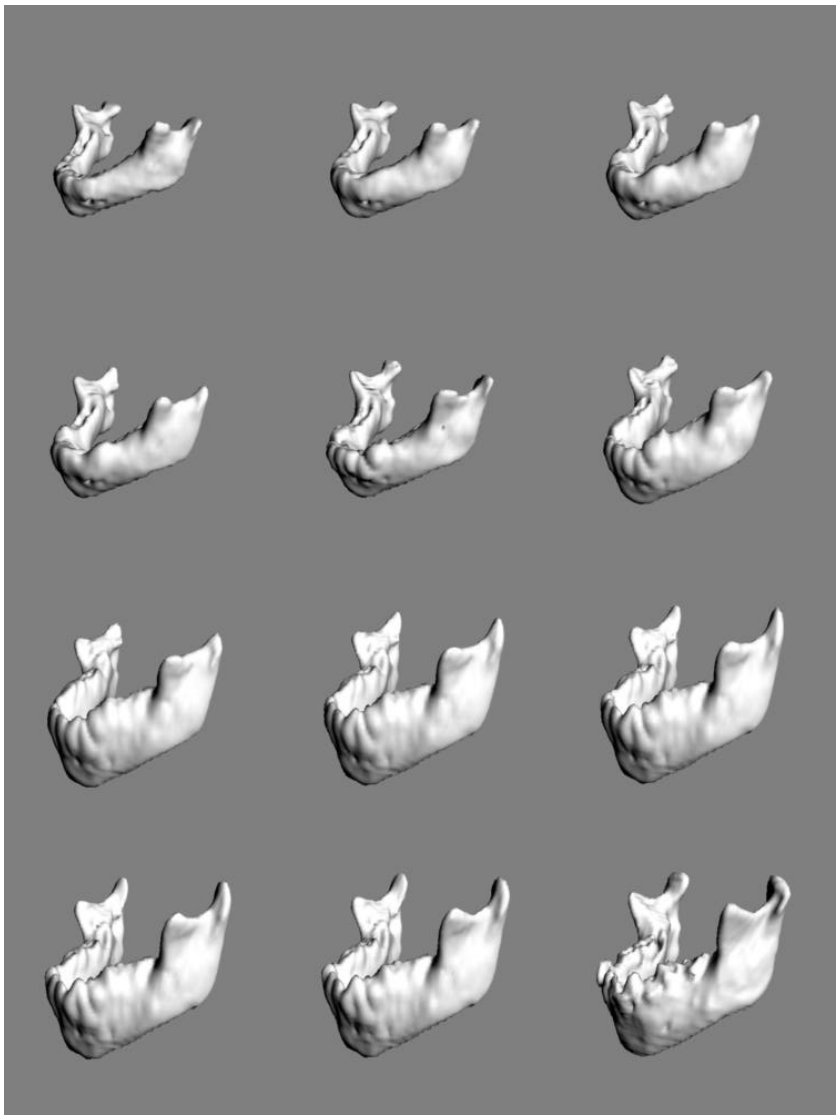

Figure 5.9: *Fluid registration of 3 binary jaw images - top left, 2. row middle, and bottom right. The other images show intermediate results.*

CT scans and binarized. The binary images have then been registered. The figure shows extracts from a movie generated from intermediate images. This application is similar to that of Christensen et al. in [Chr96b] where they register skulls of infants elasticly, or that of Subsol et al. [Sub96] who also register skulls of infants elasticly.

5.4 Comparing fluid with 'demon'-based registration

In [Thi96] Thirion proposed a 'demon'-based registration method. This is an iterative algorithm, where forces are determined in the template image based on equations that are very similar to the body force used in this work. His equation 4:

$$\tilde{\boldsymbol{f}} = \frac{-(T-S)\nabla T}{\nabla T^2 + (T-S)^2} \tag{5.18}$$

is in fact just a scaled version of the body force used here (equation 5.5).

When Thirion deforms an image to match another, he performs an iterative process where body forces are determined using equation 5.18, the force field is lowpass filtered using a Gaussian filter and finally integrated over time.

Comparing the 'demon'-based algorithm with the algorithm in this paper, we see that:

- The body forces are almost the same.
- The lowpass filtering using a Gaussian corresponds to the application of the linear elastic filter.
- The time integration of the lowpass filtered force field corresponds to the Euler integration performed using equation 5.6.

We, therefore, conclude that the approach proposed in [Thi96] is similar to the viscous fluid registration using convolution, which is proposed here. Applying the Gaussian, instead of the linear elastic filter, is faster since the Gaussian filter is separable in contrast to the linear elastic filter. In the next section the relationship between the Gaussian and the linear elastic filter is illustrated by theoretical arguments.

5.4.1 Characteristics of the Gaussian filter

The linear elastic filter works by elasticly smoothing the applied force over the domain. In this respect the linear elastic filter can be regarded as a lowpass filter like the Gaussian. On the other hand, the Gaussian filter can also be seen as a physical model. Let us illustrate this by the following theorem constructed using work by Nielsen et al.:

Theorem 5.3 *Finding the Tikhonov regularized solution $f(x)$ of the signal $g(x) \in \mathcal{L}^2(\mathbb{R})$ by minimizing the energy:*

$$E[f] = \frac{1}{2} \int \left[(f-g)^2 + \sum_{i=1}^{\infty} \lambda_i \left(\frac{\partial^i}{\partial x^i} f \right)^2 \right] dx \tag{5.19}$$

with the coefficients ($t > 0$):

$$\lambda_i = \frac{t^i}{i!} \tag{5.20}$$

is the same as convolving the signal g with a Gaussian with the Fourier transform:

$$\hat{h} = \frac{1}{\sum_{i=0}^{\infty} \frac{t^i}{i!} \omega^{2i}} = e^{-\omega^2 t} \tag{5.21}$$

Proof *Nielsen et al. [Nie94], proposition 1 and 3, and result 1.* □

This result shows that convolution with a Gaussian can be seen as finding a solution corresponding to the minimization of an energy functional, ie. a physical problem. A similar result exists for the multi-dimensional Gaussian [Nie94].

We note that the solution obtained by convolving with a Gaussian does *not* satisfy any specified boundary conditions. Remember that boundary conditions are implicitly a condition for the homeomorphic mapping of the domain, since the homeomorphic mapping requires the mapping to be onto. This was partly satisfied by requiring the boundary of the domain to be mapped onto the boundary.

In theory the solution using the Gaussian does, therefore, not strictly satisfy the homeomorphic requirement for the transformation. But, in practice it is possible to circumvent this problem by inserting the image into a larger domain, thus adding a broad boundary to the image in all directions. On the other hand this increases the computational complexity of the problem.

We state another result from the work of Nielsen et al. [Nie96]:

Theorem 5.4 *Any analytical function $f : R \to R$, that is faster than polynomially decaying, can be decomposed into a weighted sum of Gaussian derivatives $\phi^{(i)}(x, s)$ with scale parameter s:*

$$f(x) = \sum_i \gamma_i \frac{(-s)^i}{i!} H_i(\frac{x}{s}) \phi^{(i)}(x, s) \tag{5.22}$$

where H_i are the Hermite polynomials and the weights γ_i are identified by the Hermite moments:

$$\gamma_i = \int H_i(\frac{x}{s}) f(x) dx \tag{5.23}$$

Proof *Nielsen et al. [Nie96, Nie96b].* □

This theorem indirectly states, that the zero-order approximation of any function that satisfies the requirements of the theorem is the Gaussian. A similar

proof exists for the multi-dimensional case [Nie96b]. We can thus understand the Gaussian as a (very) low-order approximation of the linear elastic filter. It is, therefore, reasonable to conclude that the approach of Thirion [Thi96] is an approximation to the viscous fluid registration.

When compared to the linear elastic filter it is obvious that the Gaussian filter gives no displacement orthogonal to the direction of a force. The linear filter does this (see figure 5.3) and it is important for the viscous effect of the filter.

5.4.2 The Gaussian versus linear elastic filter for registration

To investigate the differences between the two filters, the fluid registration algorithm has been reimplemented using the Gaussian filter. Experiments using the linear elastic and Gaussian filters are shown in figures 5.10, 5.11, and 5.12.

All the results look very similar and quite acceptable, although sometimes different when the grids are observed. In particular the circle to 'C' experiment shows that the stretching of the template is not quite the same. Indeed when one looks closely on the inside of the 'C', the right side of the middle of the stretched bullseye has been compressed more with the Gaussian filter. This could indicate a topological problem for thin anatomical structures.

In addition, the patch to 'C' experiment in figure 5.10 shows that the central grid for the Gaussian filter is more dilluted than for the linear elastic filter. This is probably caused by the lack of viscous contraction, which the elastic filter introduces with the $\tilde{\boldsymbol{\nabla}}(\tilde{\boldsymbol{\nabla}} \cdot \tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}))$ term.

Whether these problems are of any concern must depend on the application.

5.5 Timings

Table 5.1 shows the results of the 2D experiments which can be compared to the similar experiments reported in [Chr94c]. To be able to compare the number of iterations for multi-resolution and single-resolution, the number of iterations for the multi-resolution work has been calculated, as a weighted sum of the iterations in each resolution. As weight are used the relative voxel distance in the resolution, since in each resolution the maximum flow distance is scaled to a factor of a voxel.

The table shows that the results of using the convolution approach on a *single workstation*, can be compared to those of Christensen et al. on a 128x64 DECmpp 12000 Sx/Model 200 massively parallel computer. These timings are stable and consistent with the other results using realistic data.

Using the Gaussian instead of the linear elastic filter gives faster results. But, it is interesting to note that more iterations are required in each resolution. This could be interpretated as a problem related to using the Gaussian filter.

When comparing our results using the linear elastic filter to estimates of timings on a MIPS R4400 processor [Chr96], it is estimated that a speedup of

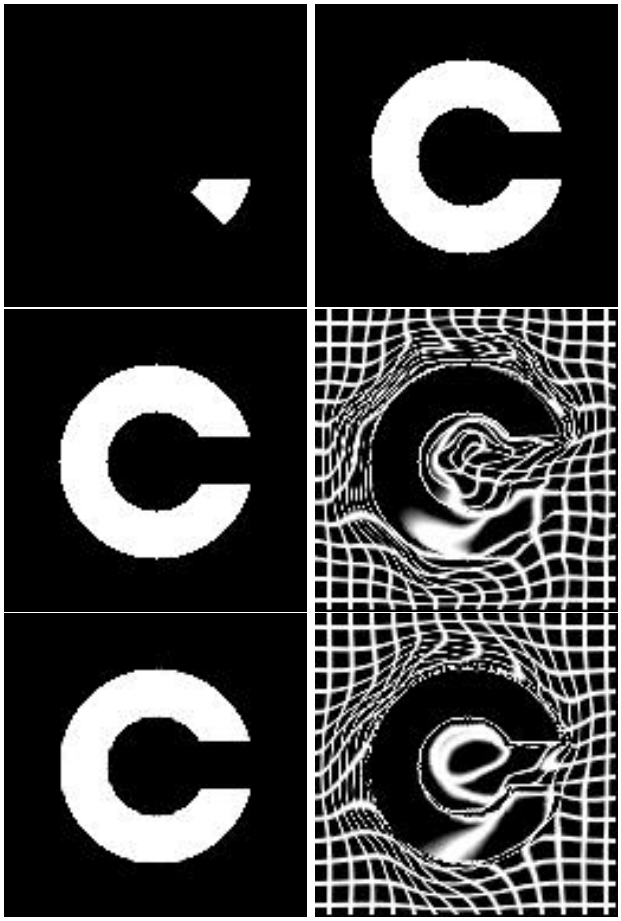

Figure 5.10: *Comparison between the linear elastic and Gaussian filters for the patch to 'C' experiment. Top: Template and study images. Middle: Result using linear elastic filter. Bottom: Result using Gaussian filter.*

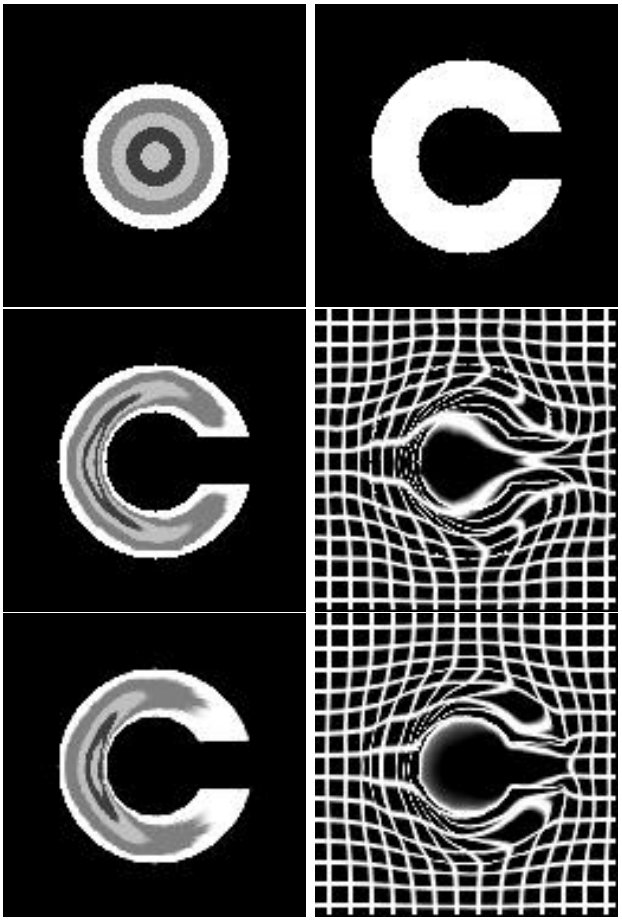

Figure 5.11: *Comparison between the linear elastic and Gaussian filters for the circle to 'C' experiment. Top: Template and study images. Middle: Result using linear elastic filter. Bottom: Result using Gaussian filter.*

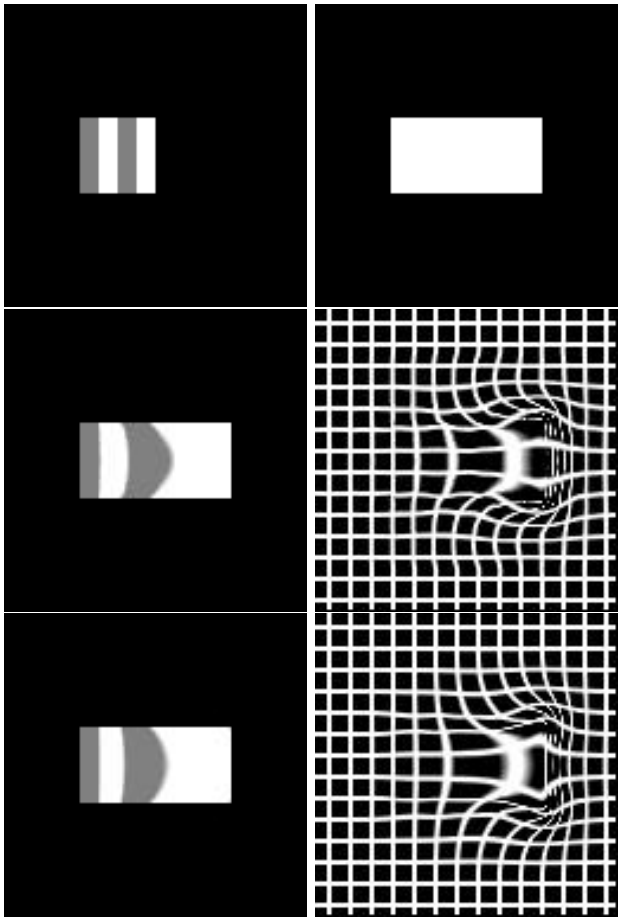

Figure 5.12: *Comparison between the linear elastic and Gaussian filters for the square to rectangle experiment. Top: Template and study images. Middle: Result using linear elastic filter. Bottom: Result using Gaussian filter.*

Table 5.1: *Timings for 2D viscous fluid registration experiments with filters compared to results using original algorithm.*

	Linear elastic filter (seconds [iter.])	Gaussian filter (seconds [iter.])	Christensen et al. [Chr94c] 128x64 MasPar (seconds [iter.])
Square - Rect. 128x128 (5.2)	37.8 [123] [20/13/17]	7.2 [138] [20/19/20]	45 [180]
Patch - 'C' 128x128 (5.10)	46.9 [408] [90/15/18]	11.3 [630] [90/90/90]	91 [250]
Circle - 'C' 128x128 (5.11)	44.2 [167] [30/13/21]	32.6 [210] [30/30/30]	423 [600]

Note: The triple numbers below the timings are the number of iterations in the three resolutions (reduction [2,1,0]). The numbers in [...] after the timings are the number of iterations. For multi-resolution the number of iterations are determined as $4n_2+2n_1+n_0$, where n_i is the number of iterations for a resolution with reduction factor i.

at least an order of magnitude is achieved for the same single processor platform.

This speedup is based on a software implementation of the convolution. Use of specialized convolution hardware, as found in eg. the RealityEngine II graphics board from SiliconGraphics, should speedup the registration even more.

5.6 Summary

This chapter has presented a new algorithm, which reduces the time consumption of the viscous fluid registration algorithm by Christensen et al. [Chr94c], with at least an order of magnitude. This speedup was achieved by implementing the core process of the registration method using convolution with a filter developed in this chapter. The results are similar to those of Christensen et al.

It was also shown that the 'demon'-based registration method by Thirion [Thi96] is similar to the viscous fluid registration method developed here. This insight comes from the implementation of the core routine as a filter. Using this filter, the numerical implementations of the two methods look very similar. With results from scale-space theory [Nie94, Nie96, Nie96b] it was shown that the Gaussian filter is indeed an approximation to the linear elastic filter. Examples showed quite similar results although some problems were indicated for the Gaussian filter.

Chapter 6

Non-Rigid Registration using Bone Growth Model

Bone growth is a biological process far more complex than immediately expected by the layman. Although, the growth is controlled by some very simple processes, the pattern of the growth can be surprisingly sophisticated.

This chapter studies the process of bone growth in the human mandible. Based on the available medical knowledge, a bone growth model is established, and experiments are carried out to determine the validity of the model. These experiments uses a unique data set containing three voxel images of the mandible of a child, taken at different times[1].

Being able to model the growth of the facial bones is both interesting for understanding the development of the cranio-facial complex, and for understanding the effects of cranio-facial surgery on children. After children with growth defects have undergone surgery to restore normal function and appearance, the bones continue to grow. This growth often introduces new deformations and new surgery has to be performed.

It is, therefore, important to understand the growth of the bones. Optimally, it would be useful for the surgeon to be able, not only to predict the instantaneous result of the surgery (as discussed in the next chapter), but also to predict the evolution of the face after surgery. Using such knowledge, the surgeon could possibly anticipate this development and correct for it in advance.

The process of modeling the growth can be seen as a registration problem, and the discussion follows the notation and semantics of the general registration methodology used in the previous chapters. Indeed the inspiration, for the development of the growth based registration algorithm, came from the experience with fluid registration of the jaw images (figure 5.9).

[1] This data set was kindly made available by Prof. Sven Kreiborg of the School of Dentistry, University of Copenhagen, Denmark.

6.1 Non-rigid registration as a physical problem

Non-rigid registration is used to register images that are different. As 'non-rigid' indicates, more than just a simple superposition of the images is required to get a complete correspondence, and some sort of deformation must be included in the transformation, to map dissimilar images onto each other.

Optimally, the non-rigidness of the transformation allow a full correspondence to be determined, and at the same time limit excessive deformations by including some sort of regularization effect.

As illustrated in the previous chapter, most non-rigid registration methods are directly or indirectly based on elastic or viscous-fluid models. Although many methods do not use the terminology of these continuum models, they follow the same ideas algorithmicly. As an example, the application of the Gaussian filter, to a driving force field, is the same as Thikonov regularization of this field [Nie94], and thus the application of a physical model similar to an elastic deformation.

The use of elastic and viscous fluid models for registration of images has often been criticized, because they represent the application of a physical model, which does not reflect the actual physical development between the images.

A correct physical model, for registration of images of the brain of two individuals, would reflect the actual biologic and genetic development of these brains. Since a common genetic basis exists for these brains, it should, theoretically, be possible to create a development model for the two brains down through history to a common ancestor. Using this development model a correspondence could be created.

The difference between the traditional registration models and the idea of a physically correct model of development, is illustrated in figure 6.1. Rigid registration methods basicly register images to derivatives of themselves. The traditional non-rigid registration methods attempt to model the difference between two images, using an *arbitrary* physical model, which does not reflect the actual physical process. The development methods, on the other hand, use the *actual* physical process as the model of change.

One of the important advantages, of using a physically correct model, is the ability to *simulate* what happens between the images. Where traditional non-rigid methods only provide a correspondence between the images, the physically valid models describe the actual development, which can then be simulated. Naturally, these models have generally been viewed as impossible, although conceptually interesting.

In this chapter a physically valid development model, for non-rigid registration of time sequence images of the human mandible is developed. With a model of physical bone growth, images are registered by allowing the initial image to grow to the second image. The initial and final images are used as boundary conditions.

A set of three images of the mandible of the same child at different times, is used to validate the growth model (see figure 6.2). These images have been acquired at the age of 9, 21 months and 7 years.

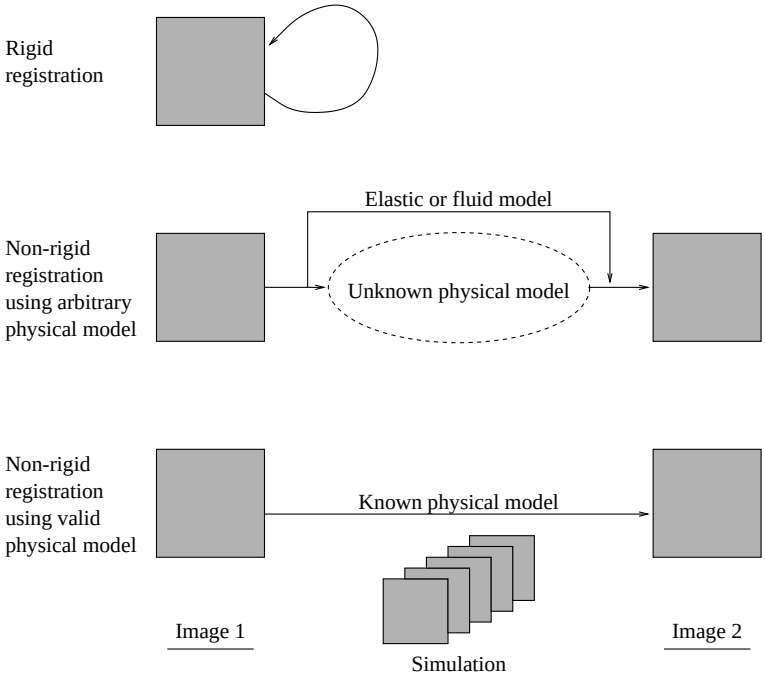

Figure 6.1: *Different registration methods and their use of physically valid or invalid models of the change between two images.*

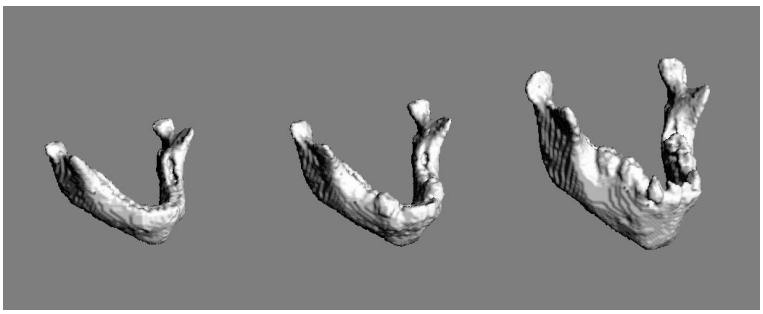

Figure 6.2: *Images of the mandible of a child at the age of 9, 21 months and 7 years.*

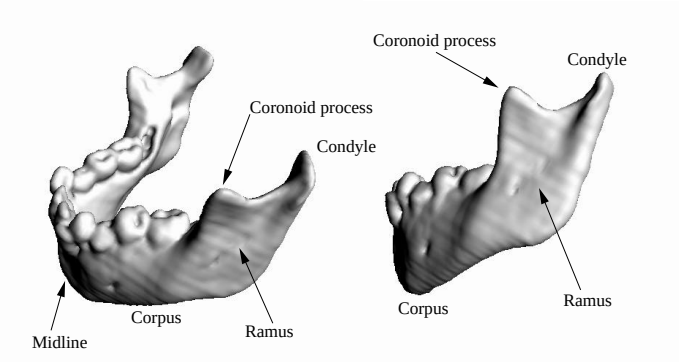

Figure 6.3: *Anatomic description of the mandible.*

During this period both actual bone growth, and teeth growth and eruption take place. But for simplicity, we ignore the teeth growth and consider the mandible as one coherent piece of bone. The teeth and other internal structures are only used for initial rigid registration of the images.

6.2 Bone growth

A general rule for bone growth, is that growth only takes place on the surface of bones. The interior of bones is calcified and rigid, and do not change shape [Goo82, Lin86]. Bone growth is consequently different from normal biological growth, where growth happens by cell division.

Bones grow by a complex process of *deposition*[2] and *resorption*. Deposition adds new material to the surface of the bone and resorption removes material from the surface of the bone. These processes only occur in a thin boundary layer, in which cells called osteoblasts and osteoclasts are responsible for the deposition and resorption of bone material respectively.

Deposition can take place both on the surface of normal bone, in which case it is called *surface deposition*, in *sutures* where bones are connected by thin bands of fibrous material, and in the *cartilages* that join some bones. Examples of sutures are found in the skull, where they accommodate the growth of the brain, and examples of cartilages are found on the condyles of the mandible, where they form the joint between the mandible and the skull.

Surface deposition is considerably slower than the other forms of growth, but takes place on the entire surface. Cartilage and suture growth only take place in isolated areas.

6.2.1 Growth of the mandible

The general anatomy of the mandible is shown in figure 6.3.

[2]Sometimes called apposition.

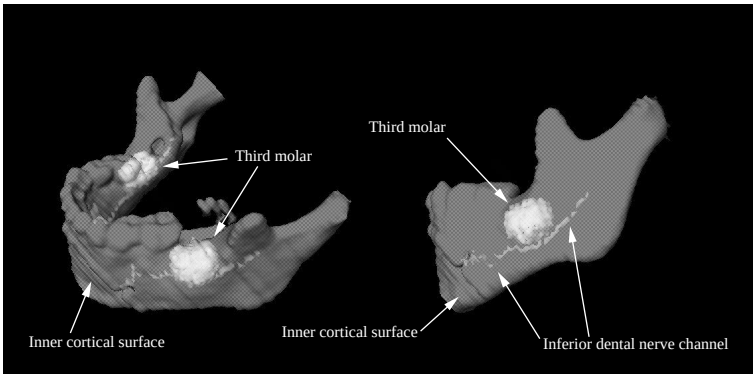

Figure 6.4: *Stable features in the mandible.*

Up to the second half of the first year the mandible has cartilage in the midline. But after this period the two halves are connected and the cartilage disappears. The mandible is then a single bone, with cartilages on the two condyles (the mandibular condylar cartilages).

Growth of the mandible takes place by growth in the condylar cartilages, and by surface deposition and resorption on the remaining surface of the mandible.

That the mandible only grows by surface processes can seem surprising, considering the large size difference between the jaw of a child and an adult. But implant studies by Björk [Bj61, Bj63] and Björk and Skieller [Bj72, Bj77] have shown this model of the growth process to be valid at least for the corpus. The growth pattern of the condyles is still under discussion.

6.2.2 Stable structures

Since the interior of the mandible does not grow, it should be expected that stable structures could be detected here.

Indeed based on his implant studies, Björk [Bj69] suggested the inferior dental nerve channel, the inner cortical surface of the symphysis, the lower margin of the third molar tooth germ before it erupts, and the tip of the chin as stable features of the mandible (see figure 6.4).

There is general agreement on these features, although the third molar is only stable during the period, in which it has not started developing roots. It can, therefore, not be used in studies over many years.

6.3 Non-rigid registration of mandibles using bone growth model

Using the information about the growth of the mandible, this section establishes a mathematical model of the growth process, and a simulation algorithm, which

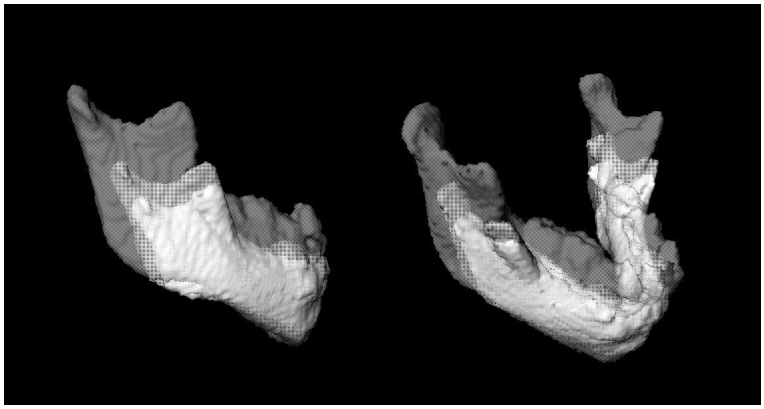

Figure 6.5: *Overlayed mandibles after rigid registration.*

allows non-rigid registration of time sequence images of a developing jaw and, thereby, simulation of it's development.

The basic algorithm first rigidly registers the mandibles using the stable structures suggested by Björk [Bj69]. When these structures have been overlaid, the remaining difference between the images reflects the growth that has taken place between their acquisition.

A partial differential equation for the growth on the surface of the bone is set up, using the estimated distance between corresponding features to determine the growth velocity.

Finally, the growth process is simulated to register the images.

6.3.1 Rigid registration of stable structures

A combination of surface based rigid registration, based on the Iterated Closest Point (ICP) algorithm by Besl and Kay [Bes92], and manual correction, were applied to register the mandibles. Using the ICP algorithm, the tip of the chin was first registered and the result subsequently validated and corrected using the other stable structures. Figure 6.5 shows the three mandibles after rigid registration.

From the overlapping mandibles it is possible to discern some of the changes caused by the growth. The most dramatic growth takes place on the ramus, which is the large structure at the end of the mandible, and the condyles, which are the attached to the rami and serve as the connection to the rest of the skull.

6.3.2 Growth model

As we saw above, the deposition and resorption processes work in a thin boundary layer on the surface of the mandibles. During the work of the osteoblast and osteclast cells, the density of the bone increases or decreases in the bound-

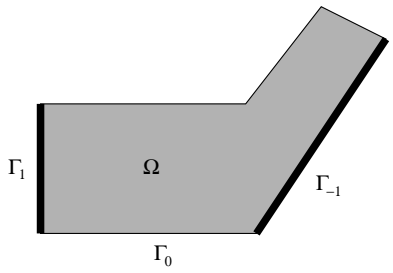

Figure 6.6: *Definition of growth domain.*

ary layer until bone has either been completely deposited or resorped. The boundary layer then moves, and the process restarts.

To model this, the bone density $\rho(\boldsymbol{x}, t) \in [0; 1]$ is used as the primary variable. Bone is defined as $\rho(\boldsymbol{x}, t) = 1$ and air is defined as $\rho(\boldsymbol{x}, t) = 0$. Thus, the definition of the bone body becomes:

$$\Omega = \{\boldsymbol{x} | \rho(\boldsymbol{x}, t) = 1\} \tag{6.1}$$

Using the same notation as in the previous chapter, we define the initial image as the *template* image $T(\boldsymbol{x}) \in [0; 1]$ and the desired end-state as the *study* image $S(\boldsymbol{x}) \in [0; 1]$. The initial domain can therefore be written as:

$$\Omega = \{\boldsymbol{x} | \rho(\boldsymbol{x}, 0) = T(\boldsymbol{x})\} \tag{6.2}$$

The registration problem is formulated as changing the density map using a model of growth, so that $\rho(\boldsymbol{x}, t_{end}) = S(\boldsymbol{x})$ where t_{end} is the time of full registration.

The boundary of the domain is denoted $\tilde{\Gamma}$ (see figure 6.6), and consists of a stable part $\tilde{\Gamma}_0$, where the template and study boundaries match, and unstable parts $\tilde{\Gamma}_1$, and $\tilde{\Gamma}_{-1}$ where deposition and resortion take place respectively:

$$\begin{aligned}
\tilde{\Gamma} &= \tilde{\Gamma}_0 + \tilde{\Gamma}_1 + \tilde{\Gamma}_{-1} \\
\tilde{\Gamma}_0 &= \{\boldsymbol{x} \in \tilde{\Gamma} \mid T(\boldsymbol{x}) = S(\boldsymbol{x})\} \\
\tilde{\Gamma}_1 &= \{\boldsymbol{x} \in \tilde{\Gamma} \mid T(\boldsymbol{x}) < S(\boldsymbol{x}) \wedge \rho(\boldsymbol{x}, t) \in [0; 1[\} \\
\tilde{\Gamma}_{-1} &= \{\boldsymbol{x} \in \tilde{\Gamma} \mid T(\boldsymbol{x}) > S(\boldsymbol{x}) \wedge \rho(\boldsymbol{x}, t) \in]0; 1]\}
\end{aligned} \tag{6.3}$$

For completeness we define the boundary as having a thickness $\delta > 0$.

We now proceed to define the partial differential equation (PDE) governing the growth process. The general requirements for this PDE is that change in density should happen only for $\boldsymbol{x} \in \tilde{\Gamma}_1 + \tilde{\Gamma}_{-1}$, and that the change in density should reflect the velocity of the actual growth. The PDE is defined as:

$$\frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} = \begin{cases} cv_g(\boldsymbol{x}, t) & x \in \tilde{\Gamma}_1 \\ -cv_g(\boldsymbol{x}, t) & x \in \tilde{\Gamma}_{-1} \\ 0 & x \in \tilde{\Gamma}_0 \cup \tilde{\Omega} \end{cases} \tag{6.4}$$

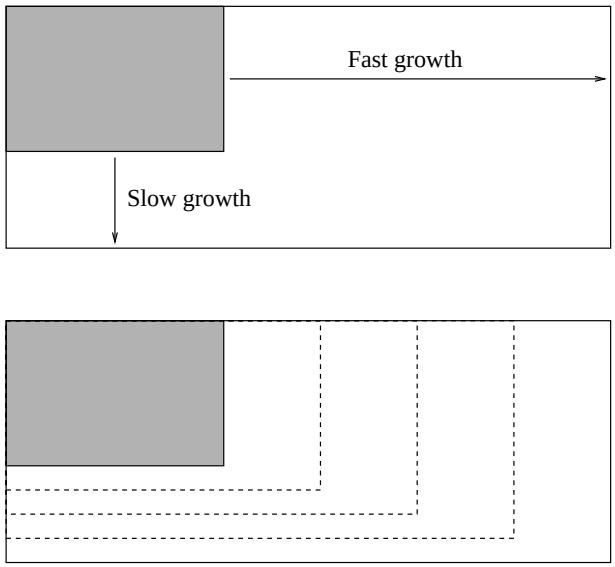

Figure 6.7: *The speed of growth must be depend on the distance to be filled.*

where $v_g(\boldsymbol{x}, t)$ is the growth velocity and c is a constant. The equation is conceptually simple, and states that the density should increase in areas of deposition, and decrease in areas of resorption. The rate of change is proportional to the growth velocity.

Determining the growth velocity

A major problem, in developing the growth registration algorithm, is the selection of an appropriate growth velocity. The initial suggestion is a constant. With a constant growth velocity, the growth process can be compared to the fluid model in the previous chapter.

Consider figure 6.7. As this drawing illustrates, for the growth to look normal, it is necessary that the growth velocity reflects the distance, which needs to be covered by the process. Otherwise, the growth would first fill out the thin spaces, and then gradually fill out the wide spaces, thus falsely indicating that the thin spaces only grow in the beginning of the process.

Instead of a constant, the actual remaining distance from the boundary of the current bone body to the boundary of the study image $S(\boldsymbol{x})$, could be used as the velocity. But this does not work either since the velocity would decrease as the gap was narrowed. The growth velocity must reflect the *initial* distance from the template image boundary to the study image boundary.

The suggested growth velocity is therefore determined using the following algorithm:

1. Initially, determine a distance map $\text{dist}(\boldsymbol{x})$ from the study image $S(\boldsymbol{x})$ to

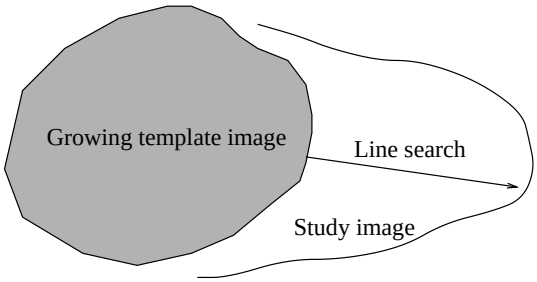

Figure 6.8: *Line process.*

the template image $T(\boldsymbol{x})$.

2. For each boundary voxel $\boldsymbol{x}_b \in \tilde{\Gamma}_1 + \tilde{\Gamma}_{-1}$ in the currently evolved image:
   (a) Perform a line search, perpendicular to the boundary starting from the current bone body $\tilde{\Omega}$, until a boundary voxel $\boldsymbol{x}_s$ in the study image $S(\boldsymbol{x})$ is found (see figure 6.8).
   (b) Set the growth velocity $v_g(\boldsymbol{x}_b, t)$ to $\text{dist}(\boldsymbol{x}_s)$.
3. Lowpass filter the velocity field using a Gaussian kernel.

The last action has been introduced to smooth discontinuities in the velocity field, and to remove spikes.

6.3.3 Implementation

The complete growth registration is implemented with the following iterative algorithm:

1. Determine a distance map $\text{dist}(\boldsymbol{x})$ from the study image $S(\boldsymbol{x})$ to the template image $T(\boldsymbol{x})$,
2. Determine the correct boundary $\tilde{\Gamma} = \tilde{\Gamma}_0 + \tilde{\Gamma}_1 + \tilde{\Gamma}_{-1}$ of the bone body $\tilde{\Omega}$.
3. For all voxels on the changing boundary $\tilde{\Gamma}_1 + \tilde{\Gamma}_{-1}$ determine the growth velocity using the algorithm above,
4. Calculate $\partial\rho/\partial t$ using equation 6.4,
5. Perform time integration of the density:
$$\rho(\boldsymbol{x}, t) = \rho(\boldsymbol{x}, t) + \triangle \frac{\partial \rho(\boldsymbol{x}, t)}{\partial t} \tag{6.5}$$
where $\triangle$ is the time step.
6. Where the density $\rho(\boldsymbol{x}, t) > 1$ or $\rho(\boldsymbol{x}, t) < 0$, disperse the excess equally to neighbouring boundary voxels.

Figure 6.9: *Initial image (left), image to grow to (middle), and the areas of differences (right).*

Figure 6.10: *Development of growth when a simple velocity model is used.*

7. If $\rho(\boldsymbol{x}, t)$ has changed, go to 2

The dispersion of excessive density in step 6 is necessary because of the final size of the time step.

6.3.4 Results

A 2D example is used to illustrate the effect of the choice of non-constant growth velocity. Figure 6.9 shows the template and study images and their differences. As can be seen in the figure, both deposition and resorption is necessary during the growth process.

Figures 6.10 and 6.11 show the growth pattern for growth with constant growth velocity (6.10) and distance dependent growth velocity (6.11). The image in the bottom right of both figures shows the development pattern. Dark values indicate early growth and bright values late growth. It is easy to see how the growth, using the distance dependent growth velocity, has a more uniform

Figure 6.11: *Development of growth when a distance dependent velocity model is used.*

development, whereas the pattern of the growth using constant velocity is more non-uniform.

The result of using the growth registration algorithm on the mandible images is shown in figure 6.12. When this figure is compared to figure 5.9 in the last chapter, the growth registration is more uniform than the fluid registration and also looks more realistic.

6.4 Summary

In this chapter a model of bone growth in the human mandible has been developed. The results provide documentation that the theories of bone growth, which the model is based upon, *may* be correct. The documentation is not a definite proof, since it only shows that the theory *can* explain the growth of the mandible.

An important result of modeling the growth, is the time images, such as those in figures 6.10 and 6.11, which are a by-product of the simulation. Together with corresponding images of the density change in each iteration, they convey important information about the structure of the growth. The clinical importance of these images has yet to be investigated.

This chapter has argued that modeling the growth process, is an image registration process. As such, using the growth model as the basic control mechanism, represents a new form of medical image registration. All previous non-rigid registration algorithms have used physical models as regularization methods, and have not modeled the actual physical development responsible for the differences between the images.

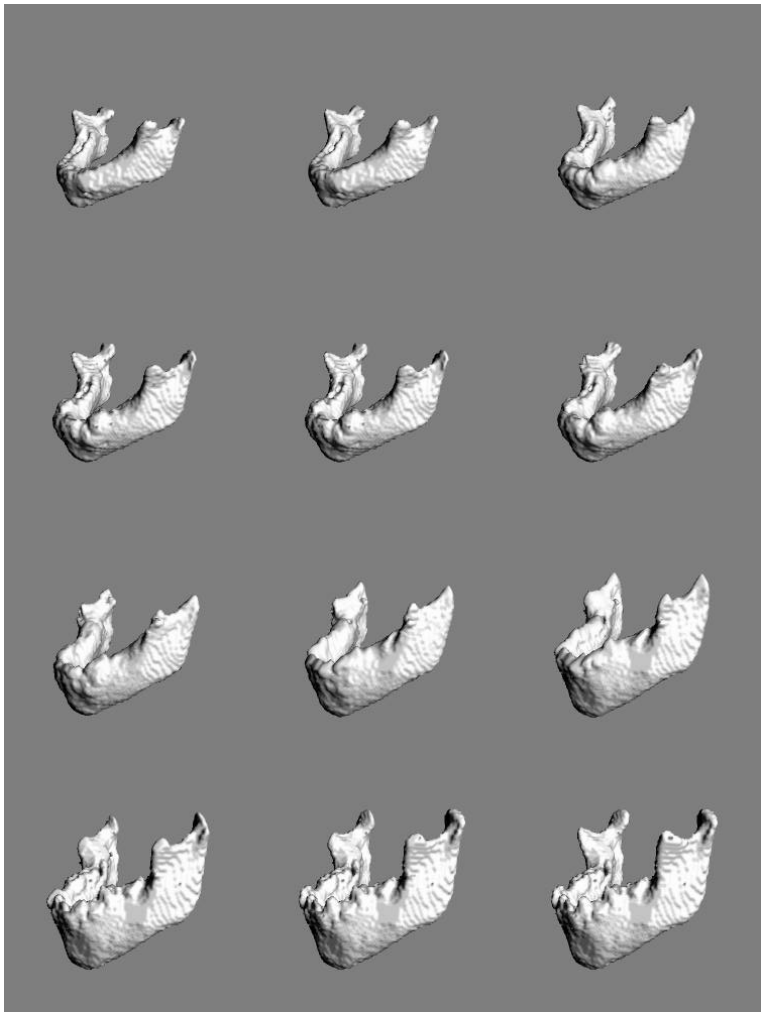

Figure 6.12: *Result of growth registration of 3 binary jaw images - top left, 2. row middle, and bottom right. The other images show intermediate results.*

Chapter 7

Soft Tissue Modeling in Surgery Simulation

Just like flight simulators are essential today in the education of aircraft pilots, surgery simulators are expected to have an important impact on the education of surgeons in the future [Sa93].

Currently training of surgeons is either performed using animals, cadavers, or actual human patients. But, it is expected that these training methods will be restricted in the future. Animal protection groups are pressuring the medical industry to restrict the use of animals for experiments, cadavers are generally problematic, and the use of direct training (under supervision) on human patients will probably increasingly be limited by insurance problems and patient scepticism. In addition, the facilities necessary for training on animals are quite expensive.

Virtual reality techniques are, therefore, being developed to replace experimentation on live objects with simulated environments. In particular, the fields of cranio-facial and minimally invasive surgery have seen the development of some initial surgery simulators.

7.1 Cranio-facial surgery simulation

Cranio-facial surgery is used to change the bone structure of a patient's face. The purpose of this kind of surgery, is to enhance the appearance of the patient or to restore normal function, when abnormal bone growth has resulted in restriction of body function or development.

During surgery, parts of the cranio-facial bones are separated from the remainder, and rearranged like building blocks, by the surgeon. Naturally, extensive planning takes place to determine the best procedure.

This planning is usually performed using x-ray cephalometric techniques and reconstructed CT images. Some cranio-facial teams are using solid modeling

software (eg. [Lo94]), but most use manual ad-hoc methods during the planning phase.

Although the surgery generally aims at improving the overall facial appearance of the patient, traditionally only the effects on the bone structures have been considered systematicly during planning. The relationship to the soft tissue surface of the patient has often been left to experience and guessing.

But, in recent years promising work has been done to develop simulation software, which can model the soft tissue changes, that are the result of changes in the underlying bone structure of the patient.

The requirements for such simulators are primarily realism, and since speed is not an important concern, most of the attention has been directed at the development of realistic soft tissue models.

These simulators are therefore not realistic models of the surgery situation, but rather software tools for modeling and planning surgery.

7.2 Minimally invasive surgery simulation

Minimally invasive procedures such as endoscopy and laparascopy are replacing more and more normal procedures, with extensive benefits for both patients and hospitals in terms of reduced hospitalization time and fewer complications.

These procedures all share the characteristic that the surgeon's interface to the patient is limited to a TV screen, and tools that are inserted into the patient through either natural body openings or small incisions. The interface is, consequently, dramatically reduced in comparison to open surgery, where the surgeon has direct contact to body tissues and fluids.

Because of the simplicity of the patient/surgeon interface, surgery simulators are relatively easier to develop for minimally invasive surgery. At the same time special motoric skills are needed for this kind of surgery, which can only be acquired by extensive practice with the actual instruments. Together these factors have driven the research into virtual reality techniques to replace the current use of either animals or simplistic 'brick and block' setups for training of surgeons.

Since virtual reality simulators require real-time response, research has concentrated on developing high-speed simplified elastic models and computer graphics.

7.3 Technical requirements

The requirements of a general surgery simulator can be characterized by the following three demands:

Computer graphics: Graphics is needed to render realistic views of the virtual surgery scene, and provide the surgeon with a *visual* illusion of reality.

Haptic interface: A haptic interface is provided through the instruments and tools which the surgeon uses to work on the virtual reality simulator. By

tracking the position of these tools and sensing their state, the computer is able to determine the actions of the surgeon in the virtual scene, and act accordingly.

In addition, the haptic interface provides the surgeon with a physical sensation of touching and sensing objects in the scene using force-feedback techniques. The haptic interface thus represents the *sense of touch* illusion of reality.

Physical modeling: Physical models provide the surgeon with a *behavioural* illusion of reality. By modeling the visco-elastic behaviour of human skin, the fluid flow of blood from a wound etc., these models ensure that the virtual scene reflects the expectations of the behaviour of the physical reality.

Computer graphics have been extensively developed over the last decades, and is today quite sufficient for surgery simulation systems. Only speed and price is still problematic, but the general development of hardware leaves no doubt that it is only a question of time before the necessary computers will be available at reasonable prices.

Haptic interface devices for minimally invasive surgery simulation can be bought commercially, and although benefits of continued development are expected, the quality is already sufficient for many purposes.

Physical models represent the biggest obstacle today in surgery simulation. In particular, the development of real-time elastic models of human soft tissue is essential for the continued evolution of surgery simulation systems.

In engineering and physics, elastic models have been developed during the last century, and the immediate idea would be to apply these models directly for surgery simulation. Unfortunately, this is only feasible for the cranio-facial simulators that were described above.

Traditional elastic mathematical models have been developed with high emphasis on precision, and solution time has only had secondary importance. Since cranio-facial surgery simulators have similar requirements, the traditional methods can be used to some degree. But for real-time simulators, such as those used for minimally invasive surgery, the demands are often contrary to those of engineering and physics models. Virtual reality surgery simulation models are being developed with emphasis on speed. For training simulators the requirement for *precise* physical models is much less than the demand for *visually satisfying* and *fast* models.

Because of the different demands of cranio-facial surgery simulation and virtual reality systems, the development currently follows two parallel tracks as illustrated in figure 7.1. Cranio-facial simulation models use relatively realistic, but time-consuming models, and virtual reality systems use very simple real-time models. It is probable that these tracks will converge in the future as more powerful computers become available.

The contribution of this thesis, in surgery simulation, is centered on the development of visually satisfying elastic models for real-time deformation of human soft tissue. This work is described in the next chapter.

7.4 Previous work

Without going into specific technical details this section reviews the most important work on soft tissue modeling in surgery simulation.

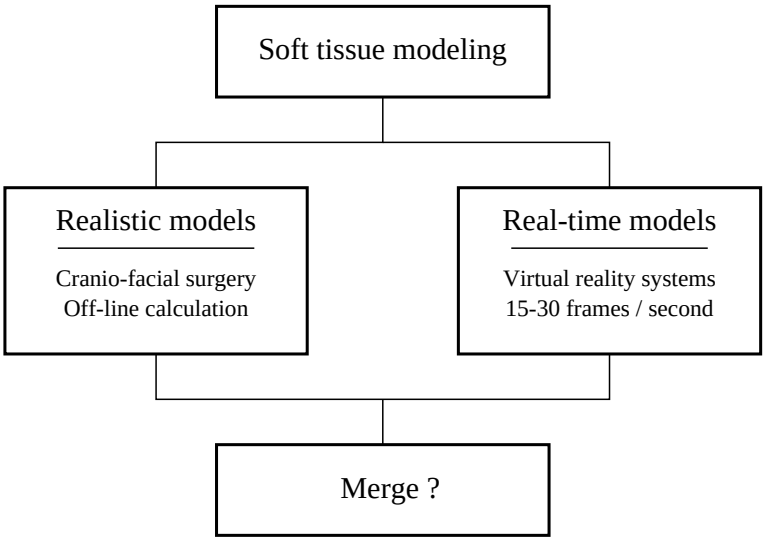

Figure 7.1: *Parallel development of soft tissue models for realistic (cranio-facial surgery) and real-time models (virtual reality training systems).*

7.4.1 Cranio-facial surgery simulation models

Early work using solid modeling techniques have been reported by Yasuda et al. [Yas90], Satoh et al. [Sat92], Caponetti et al. [Cap93], Lo et al. [Lo94], and Delingette et al. [Del94]. Delingette et al. used 2-simplex surface meshes [Del94b] to represent the surface of the skull, and also reported on some preliminary experiments using 3-simplex meshes as models of the soft tissue between the 2-simplex meshes of the skull and the skin.

In his Ph.D. thesis Pieper [Pie92, Pie92b] presented work on a complete system for simulating plastic surgery. His system used a volumetric linear elastic finite element model for the skin, and showed validation results comparing real plastic surgery results to predictions obtained using the system.

Interesting work has been reported by Waters and Terzopoulos on facial animation [Wat87, Wat91, Ter91b, Wat92, Ter93, Lee93, Lee95] in which they created non-linear mass-spring models of the facial soft tissues. By adding muscles, and modeling the muscle activity for general human expressions, they developed the ability to animate human expressions.

Keeve et al. [Kee95b, Kee95, Kee96c] have been strongly influenced by this work. They developed a similar system in which they created individualized models from CT scans and added the ability to perform surgical procedures on the bone structures using interactive computer graphics. By moving separated bone pieces and modeling the corresponding soft tissue deformation, they obtained an estimate of the resulting facial changes for the particular surgery. Some validation results have been reported.

In [Kee95b, Kee95, Kee96c] Keeve et al. used the mass-spring models orig-

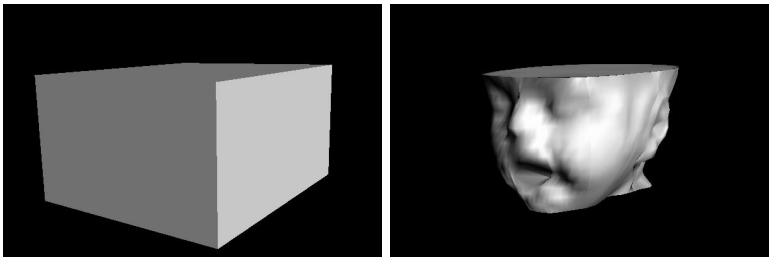

Figure 7.2: *Active cube is deformed to match the shape of a patient.*

inally proposed by Waters and Terzopoulos with slight modifications. In later work [Kee96b, Kee96] they have used finite element models of the soft tissue.

A similar system for modeling facial expressions around the mouth has also been developed by Tanaka et al. [Tan95].

Koch et al. [Koc96] have used commercially available tools to build a cranio-facial surgery simulation system. In this system, soft tissue changes are modeled using a 2D finite element surface model of the skin. The skin model is connected to the skull using springs with spring constants depending on the soft tissue type they span. The soft tissue type is determined automatically by classification of the CT scan, which provides the 3D model of the patient.

Since only a *surface* model of the soft tissue is used, the work of Keeve et al. seems more convincing, although the calculation of the spring constants based on the grey-level values of the CT scan, is an interesting new idea. Unfortunately, it is questionable whether sufficient information can be gathered from CT scans which normally are optimized for bone visualization.

In [BN95, BN95b] some preliminary work on cranio-facial surgery simulation was reported by this authour. Using a volumetric elastic model proposed earlier by Terzopoulos and Fleischer [Ter88] the deformation of the face of an individual was modeled for horizontal movement of the jaw. The elastic model can be shown to be equivalent to a St. Venant Kirchhoff hyper-elastic continuum model.

The procedure is illustrated in figures 7.2 and 7.3. First an *active cube* [BN94] is deformed to match the surface of the patient (figure 7.2). The active cube is a volumetric model and the deformed cube provides a curvi-linear parametrization of the patient tissue. Using simple brute-force techniques, the deformation of the soft tissue is then calculated when bone structures in the patient are moved (figure 7.3).

This work was discontinued in favour of the work on real-time models reported in the next chapter. Recently, M.Sc. Bo Rasmussen has implemented work similar to that of Keeve et al. during his M.Sc. thesis project, on the initiative and under the supervision of this authour [Ras96].

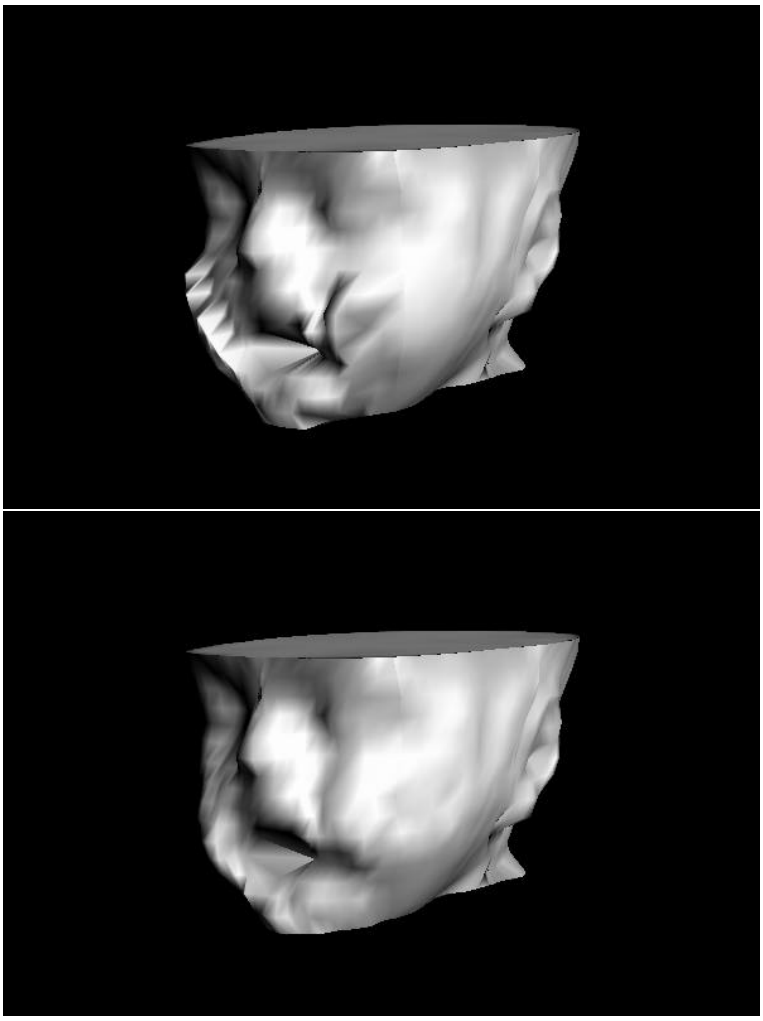

Figure 7.3: *The jaw of the patient model is moved forward 1 cm (top). After modeling the corresponding deformation of the soft tissue, the estimated shape of the 'operated' patient is shown (bottom).*

7.4.2 Real-time surgery simulation models

The demands of real-time performance have forced most researchers to develop or adapt very simplistic models of elastic deformation to the needs of surgery simulation. Eg. instead of using implicit solution methods which require the solution of matrix systems, explicit models (aka. mass-spring models) have frequently been used. Although these models suffer from poor precision and stability problems, they are very easy to implement and yield reasonable speeds. In addition they can be easily implemented on parallel processors.

Cover et al. [Cov93] were the first to present real-time models for surgery simulation. They used a simple surface-based mass-spring model to simulate deformation of a gall-bladder.

Kühnapfel, Deussen, and Kuhn [Kue93, Kue94, Kue95, Deu95, Ku96] have implemented mass-spring models in their KISMET simulation system. Although the models basicly are surface models, they introduce volumetric behaviour by including *parent* nodes which connect nodes on different sides of an object. Interesting work on determining the optimal positioning of nodes and their masses were presented in [Deu95] in which they used simulated annealing to find the optimal configuration. The resulting system, eg. as presented in [Ku96], looks quite convincing.

Surface models are also used in the commercial Teleos software developed by HT Medical, Inc., Maryland, USA[1]. They use tubular spline surface models and can only model simple structures derived from the tubular topology (eg. arteries, gall-bladder). On the other hand this system is the first commercial simulation toolkit, and functional additions can probably be expected in the future.

Implicitly solved finite element systems have been used in the on-going parallel work of Bro-Nielsen [BN95b, Cot96, BN96c, BN96d] and Cotin et al. [BN95b, Cot96, BN96c, Cot96]. The first work on the use of implicit finite element models of linear elastic models was presented in [BN95b, Cot96]. The next chapter describes later work by Bro-Nielsen [BN96c, BN96d] where a technique called condensation was introduced to reduce the complexity of volumetric finite element models. The resulting models are called Fast Finite Element (FFE) models.

The most advanced use of finite elements has been presented by Sagar et al. [Sag94]. Their eye surgery simulator used a non-linear incompressible Mooney-Rivlin elastic model with finite elements to model deformation of the eye. The finite element solution to this kind of model is normally computationally expensive, and it is currently only in such very specialized simulators it is possible to use these models.

7.4.3 Work for the future

Just as line drawing were replaced by pixel drawing for 2D graphics, today volume rendering is used more and more instead of polygonal models for visu-

[1] Previously known as High Techsplanations, Inc.

alization of 3D volume data.

All the simulators described above basicly use polygonal models such as triangle surfaces or tetrahedral finite element meshes. It is thus an interesting thought that these polygonal models one day might be replaced by volumetric voxel models.

In an eye-opening paper [Gib95], Gibson suggested the use of voxel-based models for deformable models. Gibson demonstrated how intuitively simple such tasks as collision detection become in this environment.

Unfortunately, the stumbling block for development of such models is the rendering speed which follows that of traditional volume rendering. Some speed-up can be achieved by only rendering objects in the scene that change shape or position, but the low speed is still prohibitive.

In addition, deformation of volumetric models is a considerable problem. But in contrast to the normal finite element mesh models, the size of the mesh elements is equal to the sampling size of the voxel grid, and techniques, such as those developed in chapter 5 for viscous-fluid motion modeling, could probably be used for solving the voxel-mesh deformation problem.

Although Gibson only presented 2D deformation experiments, the basic idea of using voxel-based models seems very promising.

7.5 Summary

In this chapter the basic lines of research into surgery simulation systems have been presented. The technical problems were discussed and previous work was described. The next chapter presents the contribution of this thesis to surgery simulation models in the form of Fast Finite Element models.

Chapter 8

Real-time Deformable Models for Surgery Simulation

The main problem in real-time surgery simulation, is modeling the deformation of solid volumetric objects. These objects are typicly human organs or tissues, which often have quite irregular shape and complex visco-elastic behaviour.

Some of the previous work have used *surface* models as the basic modeling method [Cov93, HT95], and realistic tissue models have only been applied in very specialized simulators, such as the eye-surgery simulator of Sagar et al. [Sag94].

The problem with surface models, is that they do not behave like volumetric 3D objects, and lack a defined interior. The first problem is purely visual, but the second is a serious limitation to the possible applications. The surgeon cannot cut a virtual organ, modeled using a surface model, since there is nothing inside the surface. Some simple cuts can be modeled, such as cutting an artery or other thin structures, but general surgical incisions are impossible.

Attention is, therefore, turning towards solid *volumetric* models, which model the complete 3D volumetric behaviour of the objects [BN95, BN95b, BN96c, BN96d, Cot96, Cot96b, Deu95, Ku96, Kue93, Kue94, Kue95]. Unfortunately the complexity of the models rises dramaticly when volumetric models are used, and real-time performance is difficult to achieve.

Another issue is the way elastic behaviour is simulated. Simulation can be performed using global parametric models [HT95, Ter91] or local mesh based models [BN95, BN95b, BN96c, BN96d, Cot96, Cot96b, Deu95, Ku96, Kue93, Kue94, Kue95] (eg. Finite Element models). To accommodate surgical incisions demands mesh based models are necessary. Only simplistic cuts can be achieved with global parametric models. Mesh based models are therefore used in this work.

This chapter discusses real-time simulation of deformable objects using 3D

volumetric Finite Element (FE) models, which result in linear matrix systems. The theory behind 3D FE models of linear elasticity is presented and various aspects of these models discussed. In particular three new improvements over previous applications of FE are suggested.

The first improvement is called *condensation*. In practice this allows us to compress the linear matrix system, resulting from the *volumetric* FE model, to a system with the same size as a FE *surface* model of the same object. Although it has the same size as the surface model, it still models the volumetric behaviour of the object.

The second improvement, concerns the way the linear matrix system is used for simulation. In contrast to the normal approach, and against the advice of finite element experts, the system matrix is explicitly inverted, and matrix vector multiplication is used to achieve a very low calculation time.

The third improvement is called *selective matrix vector multiplication*. By exploiting the sparse structure of the force vector, additional time is saved by only using the non-zero elements of the vector.

In addition, the simulation system, which has been developed for surgery simulation, is described. With this system it is possible to simulate volumetric deformation of relatively large objects with video frame-rates, ie. 15-20 frames/second.

8.1 Choice of model

The choice of a particular model must depend on the requirements of the application. A number of requirements that the model should fulfil are, therefore, formulated:

1. Deformation should be calculated in the smallest amount of time possible.

2. We do not care about the time taken for pre-calculation such as setting up equations, inverting matrices etc. If something takes 24 hours in the pre-calculation stage, but will save 0.01 second in the simulation stage, it should be done.

3. The elastic model should be *visually* convincing. The model may be physically incorrect, if it looks right.

4. In the long run it should be possible to make cuts in the model to accommodate surgical procedures. This involves changing the topology of the model, and most importantly requires models that have defined interiors, ie. volumetric models.

In particular the last requirement motivates the choice of mesh-based 3D Finite Element (FE) models. The alternative would be parametric models such as [Cov93, HT95, Ter91]. But these models do not provide the needed freedom to perform topology changes to allow cutting. Although some of the models can

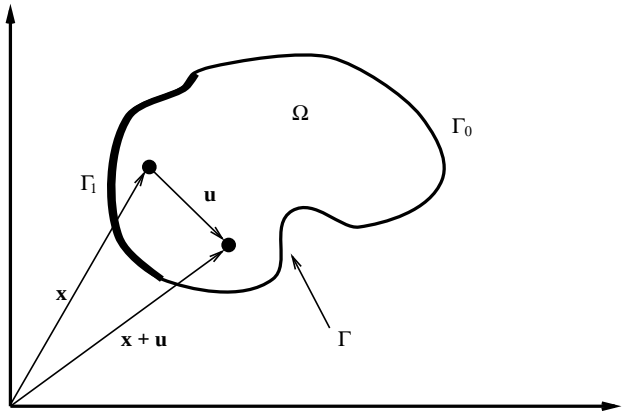

Figure 8.1: *Solid elastic object.*

handle simple cuts, the aim is to be able to make completely general cuts in the models. Only mesh-based models appear to allow this.

To meet the first two requirements, the linear elastic deformation model (see section 4.4.3) is chosen. Using linear elasticity as the basic model involves a number of assumptions regarding the physical material that is modeled. Most importantly, linear elastic models are only valid for very small deformations and strains. They are typically correct for such rigid structures as metal beams, buildings etc. Although they are used extensively in modeling, the visual result of large deformation modeling using linear elasticity is not always satisfying.

But when used with FE, these models lead to linear matrix systems which are fast, and easy to solve . There is, therefore, a trade-off between the speed of the system and the visual deformation result.

Linear elastic models are used here, because modeling general elastic volumetric deformation using FE is only *just* possible with todays computers. With faster computers in the future, more realistic models, such as incompressible Mooney-Rivlin material models [Cia87], can be expected.

8.2 Linear elastic material model

Following chapter 4 we define the elastic solid Ω as the positions $\boldsymbol{x} = [x, y, z]^T$ where $\boldsymbol{x} \in \Omega$ (see figure 8.1). The displacement of particle $\boldsymbol{x}$ is defined as $\boldsymbol{u}(\boldsymbol{x}) = [u, v, w]^T$ so that the particle $\boldsymbol{x}$ is moved by the deformation to $\boldsymbol{x} + \boldsymbol{u}$. The boundary is $\Gamma = \Gamma_0 + \Gamma_1$ where Γ_0 has fixed displacements $\boldsymbol{u}(\boldsymbol{x}) = \boldsymbol{u}_0(\boldsymbol{x})$ and forces $\boldsymbol{g}(\boldsymbol{x})$ are applied to Γ_1.

The *strain* energy of a linear elastic body Ω is defined as (equation 4.61):

$$\mathrm{E}_{strain} = \frac{1}{2} \int_{\Omega} \boldsymbol{\epsilon}^T \boldsymbol{\sigma} \, d\boldsymbol{x} \tag{8.1}$$

where the engineering strain vector $\boldsymbol{\epsilon} = [\epsilon_x\ \epsilon_y\ \epsilon_z\ \gamma_{xy}\ \gamma_{xz}\ \gamma_{yz}]^T$ consists of:

$$\begin{aligned} \epsilon_x &= \frac{\delta u}{\delta x} \qquad \epsilon_y = \frac{\delta u}{\delta y} \qquad \epsilon_z = \frac{\delta u}{\delta z} \\ \gamma_{xy} &= \frac{\delta u}{\delta y} + \frac{\delta v}{\delta x} \qquad \gamma_{xz} = \frac{\delta u}{\delta z} + \frac{\delta w}{\delta x} \qquad \gamma_{yz} = \frac{\delta v}{\delta z} + \frac{\delta w}{\delta y} \end{aligned} \tag{8.2}$$

We can rewrite this as $\boldsymbol{\epsilon} = \boldsymbol{B}\boldsymbol{u}$ where

$$\boldsymbol{B} = \begin{bmatrix} \frac{\delta}{\delta x} & 0 & 0 \\ 0 & \frac{\delta}{\delta y} & 0 \\ 0 & 0 & \frac{\delta}{\delta z} \\ \frac{\delta}{\delta y} & \frac{\delta}{\delta x} & 0 \\ \frac{\delta}{\delta z} & 0 & \frac{\delta}{\delta x} \\ 0 & \frac{\delta}{\delta z} & \frac{\delta}{\delta y} \end{bmatrix} \tag{8.3}$$

The *stress* vector $\boldsymbol{\sigma}$ is related to the strain vector through Hooke's law (see section 4.4.3):

$$\boldsymbol{\sigma} = \boldsymbol{M}\boldsymbol{\epsilon} \tag{8.4}$$

where $\boldsymbol{M}$ is the *material matrix*. For a *homogenous and isotropic* material this matrix is defined by the two Lamé material constants λ and μ:

$$\boldsymbol{M} = \begin{bmatrix} \lambda+2\mu & \lambda & \lambda & 0 & 0 & 0 \\ \lambda & \lambda+2\mu & \lambda & 0 & 0 & 0 \\ \lambda & \lambda & \lambda+2\mu & 0 & 0 & 0 \\ 0 & 0 & 0 & \mu & 0 & 0 \\ 0 & 0 & 0 & 0 & \mu & 0 \\ 0 & 0 & 0 & 0 & 0 & \mu \end{bmatrix} \tag{8.5}$$

Using these relations we can now rewrite the strain energy, and add work done by internal and external forces $\boldsymbol{f}$ and $\boldsymbol{g}$ respectively, to yield the total energy function:

$$\mathrm{E}(\boldsymbol{u}) = \frac{1}{2}\int_\Omega \boldsymbol{u}^T\boldsymbol{B}^T\boldsymbol{M}\boldsymbol{B}\boldsymbol{u}\,d\boldsymbol{x} - \int_\Omega \boldsymbol{f}^T\boldsymbol{u}\,d\boldsymbol{x} - \int_{\Gamma_1} \boldsymbol{g}^T\boldsymbol{u}\,d\boldsymbol{a} \tag{8.6}$$

where Γ_1 is the part of the surface $\Gamma = \Gamma_0 + \Gamma_1$ where external forces are applied. Fixed displacements $\boldsymbol{u}(x,y,z) = \boldsymbol{u}^0(x,y,z)$ are applied to Γ_0.

8.3 Discretization using FE model

We assume that the domain Ω of the volumetric solid has been discretized into a number of finite elements Ω^e in the form of tetrahedrons, and nodes P_q defined by $\boldsymbol{x}_q = [x_q, y_q, z_q]^T$ (see figure 8.2). The deformation at each node is specified by the deformation vector $\boldsymbol{u}_q = [u_q, v_q, w_q]^T$. In addition we also stack the displacement vectors into a compound vector:

$$\underline{\boldsymbol{u}} = \left[\ \boldsymbol{u}_1^T,\ \ \boldsymbol{u}_2^T,\ \ \ldots,\ \ \boldsymbol{u}_n^T\ \right]^T$$

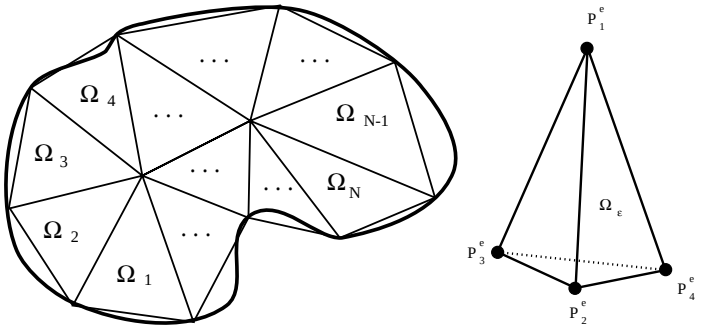

Figure 8.2: *Left: Discretization of the domain into finite elements (2D illustration) Right: Tetrahedral finite element.*

The nodes of each finite element Ω^e are denoted P_i^e, where i is the local number of the node, which is unrelated to the global numbering of the nodes. The discretized element factors are denoted with a superscripted ^{e}. The compound element displacement vector is:

$$\underline{\boldsymbol{u}}^e = \left[\begin{array}{cccc} \boldsymbol{u}_1^{eT}, & \boldsymbol{u}_2^{eT}, & \boldsymbol{u}_3^{eT}, & \boldsymbol{u}_4^{eT} \end{array} \right]^T \tag{8.7}$$

As finite elements, four-node tetrahedrons are used with linear interpolation of the displacement field between the nodes:

$$\boldsymbol{u}(\boldsymbol{x}) = \sum_{i=1}^{4} N_i^e(\boldsymbol{x}) \boldsymbol{u}_i^e \tag{8.8}$$

The basis functions $N_i^e(\boldsymbol{x})$ are defined as the natural coordinates L_i of the tetrahedron:

$$N_i^e(\boldsymbol{x}) = L_i = \frac{1}{6V^e}(a_i^e + b_i^e x + c_i^e y + d_i^e z), i = 1, 2, 3, 4 \tag{8.9}$$

The natural coordinates, the volume V^e and the coefficients $a_i^e, b_i^e, c_i^e, d_i^e$ are all defined in appendix C.

Using the fact that:

$$\begin{aligned} \frac{\partial \boldsymbol{u}}{\partial x} &= \sum_{i=1}^{4} \frac{\partial N_i^e(\boldsymbol{x})}{\partial x} \boldsymbol{u}_i^e \\ \frac{\partial \boldsymbol{u}}{\partial y} &= \sum_{i=1}^{4} \frac{\partial N_i^e(\boldsymbol{x})}{\partial y} \boldsymbol{u}_i^e \\ \frac{\partial \boldsymbol{u}}{\partial z} &= \sum_{i=1}^{4} \frac{\partial N_i^e(\boldsymbol{x})}{\partial z} \boldsymbol{u}_i^e \end{aligned} \tag{8.10}$$

where

$$\begin{aligned} \partial N_i^e(\boldsymbol{x})/\partial x &= \frac{b_i^e}{6V^e} \\ \partial N_i^e(\boldsymbol{x})/\partial y &= \frac{c_i^e}{6V^e} \\ \partial N_i^e(\boldsymbol{x})/\partial z &= \frac{d_i^e}{6V^e} \end{aligned} \tag{8.11}$$

the discretized strain energy may be written as:

$$\mathrm{E}_{strain}(\underline{\boldsymbol{u}}) = \frac{1}{2}\sum_e \int_{\Omega^e} \underline{\boldsymbol{u}}^{eT} \boldsymbol{B}^{eT} \boldsymbol{M} \boldsymbol{B}^e \underline{\boldsymbol{u}}^e d\boldsymbol{x} \tag{8.12}$$

where $\boldsymbol{B}^e$ is defined in appendix C.

The solution to the deformation problem is found when the total energy of the system assumes its minimum value. This happens when the first variation of the functional E vanishes, ie. when $\delta\mathrm{E}(\boldsymbol{u}) = 0$.

The equilibrium equation $\delta\mathrm{E}(\boldsymbol{u}) = 0$ can be split into element contributions:

$$\delta\mathrm{E}(\boldsymbol{u}) = \sum_e \delta\mathrm{E}^e(\boldsymbol{u}) = 0 \tag{8.13}$$

where

$$\delta\mathrm{E}^e(\boldsymbol{u}) = \sum_{i=1}^4 \frac{\partial \mathrm{E}^e}{\partial u_i^e}\delta u_i^e + \sum_{i=1}^4 \frac{\partial \mathrm{E}^e}{\partial v_i^e}\delta v_i^e + \sum_{i=1}^4 \frac{\partial \mathrm{E}^e}{\partial w_i^e}\delta w_i^e \tag{8.14}$$

But since the u_i^e, v_i^e and w_i^e are independent this means that

$$\frac{\partial \mathrm{E}^e}{\partial u_i^e} = \frac{\partial \mathrm{E}^e}{\partial v_i^e} = \frac{\partial \mathrm{E}^e}{\partial w_i^e} = 0,\ i = 1,2,3,4 \tag{8.15}$$

The resulting equilibrium equation for each element thus becomes:

$$\boldsymbol{o} = \int_{\Omega^e} \boldsymbol{B}^{eT} \boldsymbol{M} \boldsymbol{B}^e \underline{\boldsymbol{u}}^e d\boldsymbol{x} - \underline{\boldsymbol{f}}^e \tag{8.16}$$

where $\underline{\boldsymbol{f}}^e$ is a generalized discretized element compound force vector [1].

Because everything inside the integration sign is constant, the equilibrium equation for the finite element becomes a linear matrix equation:

$$\boldsymbol{K}^e \underline{\boldsymbol{u}}^e = \underline{\boldsymbol{f}}^e \tag{8.17}$$

where

$$\boldsymbol{K}^e = \boldsymbol{B}^{eT} \boldsymbol{M} \boldsymbol{B}^e V^e \tag{8.18}$$

is called the *stiffness* matrix [2] and V^e is the volume of the tetrahedron (see appendix C).

[1] $\underline{\boldsymbol{u}}^e$ is 12×1, $\underline{\boldsymbol{f}}^e$ is 12×1 and $\boldsymbol{B}^e$ is 6×12

[2] $\boldsymbol{K}^e$ is 12×12

The only remaining step is assembly of the global stiffness matrix and force vector from the element stiffness matrices and force vectors:

$$\begin{aligned} \boldsymbol{K} &= \sum_e \text{global}(\boldsymbol{K}^e) \\ \underline{\boldsymbol{f}} &= \sum_e \text{global}(\underline{\boldsymbol{f}}^e) \end{aligned} \tag{8.19}$$

global() is a transfer function from element node numbers to global node numbers. The result is a large sparse linear system $\boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}}$.

8.3.1 Fixing nodes

Fixing nodes means clamping them to a fixed position, as a tree is clamped to the ground at the base. If no nodes are clamped it is impossible to solve the linear system $\boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}}$ because $\boldsymbol{K}$ is singular. At least 3 nodes must be fixed for the system to be well-formed.

When nodes are assigned a fixed displacement, it changes the stiffness matrix $\boldsymbol{K}$. This change can be applied both on the element stiffness matrices and the global stiffness matrix. Only the approach for the global matrix is shown here, but the same procedure can be applied directly to the element matrices.

Without loss of generality let us assume that the displacement of the last node P_N is to be fixed as $\boldsymbol{u}_N = \boldsymbol{u}_N^0 = [u_N^0\ v_N^0\ w_N^0]^T$. Since the procedure can be seen as three identical operations modifying rows/columns $3N-2$, $3N-1$ and $3N$ of the linear system, only the modification of last row/column $3N$ is described. The modified linear system becomes:

$$\begin{aligned} &\begin{bmatrix} k_{1,1} & \dots & k_{1,3N-1} & 0 \\ \vdots & \vdots & \vdots & \vdots \\ k_{3N-1,1} & \dots & k_{3N-1,3N-1} & 0 \\ 0 & \dots & 0 & 1 \end{bmatrix} \underline{\boldsymbol{u}} \\ &= \begin{bmatrix} f_1 - w_N^0 k_{1,3N} \\ \vdots \\ f_{3N-1} - w_N^0 k_{3N-1,3N} \\ w_N^0 \end{bmatrix} \end{aligned} \tag{8.20}$$

The stiffness matrix is modified by setting the columns and rows corresponding to the elements of node N to zero with ones in the diagonal. The force vector is modified to reflect the fixed displacement of node N.

8.4 Simplifying the system using condensation

The linear matrix system $\boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}}$ models the behaviour of the *volumetric* object. This includes both surface nodes as well as the internal nodes of the model.

But for simulation purposes we are usually only interested in the behaviour of the surface nodes since these are the only *visible* nodes.

Fortunately, it is possible to remove the internal nodes from the matrix equation using a technique called *condensation* [Kar87]. The matrix equation for the condensed problem has the same size as would result from a FE *surface* model. But, it is important to understand that it shows *exactly* the same behaviour for the surface nodes as the original *volumetric* system.

Let us assume that the nodes of the FE model have been ordered with the surface nodes first, followed by the internal nodes. Using this ordering the linear system can be rewritten as a block matrix system (*s*urface / *i*nternal):

$$\begin{bmatrix} \boldsymbol{K}_{ss} & \boldsymbol{K}_{si} \\ \boldsymbol{K}_{is} & \boldsymbol{K}_{ii} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}_s \\ \underline{\boldsymbol{u}}_i \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}_s \\ \underline{\boldsymbol{f}}_i \end{bmatrix} \tag{8.21}$$

From this block matrix system a new linear matrix system may be derived, which only involves the variables of the *surface* nodes:

$$\breve{\boldsymbol{K}}_{ss}\underline{\boldsymbol{u}}_s = \breve{\underline{\boldsymbol{f}}}_s \tag{8.22}$$

where

$$\begin{aligned} \breve{\boldsymbol{K}}_{ss} &= \boldsymbol{K}_{ss} - \boldsymbol{K}_{si}\boldsymbol{K}_{ii}^{-1}\boldsymbol{K}_{is} \\ \breve{\underline{\boldsymbol{f}}}_s &= \underline{\boldsymbol{f}}_s - \boldsymbol{K}_{si}\boldsymbol{K}_{ii}^{-1}\underline{\boldsymbol{f}}_i \end{aligned} \tag{8.23}$$

The displacement of the internal nodes can still be calculated using:

$$\underline{\boldsymbol{u}}_i = \boldsymbol{K}_{ii}^{-1}(\underline{\boldsymbol{f}}_i - \boldsymbol{K}_{is}\underline{\boldsymbol{u}}_s) \tag{8.24}$$

Note, that if no forces are applied to internal nodes, $\breve{\underline{\boldsymbol{f}}}_s = \underline{\boldsymbol{f}}_s$.

Generally the new stiffness matrix will be dense compared to the sparse structure of the original system. But, since the system is to be solved by inverting the stiffness matrix in the pre-calculation stage, this is not important.

8.5 Solving the linear matrix system

Implicit solution of a linear system is often performed using iterated algorithms such as the Conjugate Gradient (CG) [Brt95]. This algorithm performs a sparse matrix vector multiplication, three vector updates and two inner products repeatedly in an iterative loop. The complexity can therefore roughly be seen as $n \times t_{Mv+3v+2vv}$, where $t_{Mv+3v+2vv}$ is the time required for the operations in one iteration and n is the number of iterations. n is seldom less than 5-10 and cannot in practice be predicted. Especially for a system where the response must come at specific frame rates, unpredictable solution time is very unfortunate.

The alternative, which is used here, is to explicitly invert the stiffness matrix. Normally this is never done when linear systems resulting from FE models are solved. The precision of the result suffers from numerical errors, and the amount

of storage needed to store a dense inverted stiffness matrix is huge, compared to the sparse stiffness matrix itself. But, as indicated in the beginning of the chapter, precision and memory size are not important when compared with speed in a trade-off.

Although the time for inversion is considerable, the solution time is very small, since it only involves a dense matrix vector multiplication:

$$\underline{\boldsymbol{u}} = \boldsymbol{K}^{-1}\underline{\boldsymbol{f}} \tag{8.25}$$

Numerical tests have been performed using the Meschach library [Ste92] to solve a linear system generated by a FE model. These experiments included explicit inversion, CG with and without preconditioner, Gauss elimination and several factorization techniques such as QR and Cholesky. When the pre-calculation time was ignored, solution by matrix vector multiplication with the inverted stiffness matrix, was at least 10 times faster than any other method.

We have not stated the actual numerical results here, since the implementation of the different algorithms in the Meschach library has not been properly optimized. The timings could therefore be different for other implementations, although the general result would be the same.

8.6 Simulation

In this section we will discuss the different simulation methods available to us. We have two linear matrix equations. One with all the nodes of the FE model and a sparse stiffness matrix. And a reduced version with only the surface nodes and a dense stiffness matrix.

8.6.1 Dynamic system

In order to use a physically correct model of the body, mass and damping are added to the static model. This is done by formulating the equation for damped harmonic motion (discrete version of equation 4.58):

$$\boldsymbol{R}\ddot{\underline{\boldsymbol{u}}} + \boldsymbol{C}\dot{\underline{\boldsymbol{u}}} + \boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}} \tag{8.26}$$

where $\boldsymbol{R}$ is the mass, $\boldsymbol{C}$ is the damping and $\boldsymbol{K}$ is the stiffness matrix of the system. $\boldsymbol{K}$ is calculated as shown above. Assuming lumped masses at the nodes, we use diagonal damping and mass element matrices:

$$\boldsymbol{R}^e_{ii} = \frac{1}{3}\rho V^e \qquad \boldsymbol{C}^e_{ii} = \alpha \boldsymbol{R}^e_{ii} \tag{8.27}$$

where ρ is the mass-density and α is a scaling factor.

The global element matrices are assembled into global matrices. Since the mass and damping matrices are diagonal, they are also block diagonal. The damped harmonic motion equation for the reduced system, therefore, simply becomes:

$$\boldsymbol{R}_{ss}\ddot{\underline{\boldsymbol{u}}}_s + \boldsymbol{C}_{ss}\dot{\underline{\boldsymbol{u}}}_s + \breve{\boldsymbol{K}}_{ss}\underline{\boldsymbol{u}}_s = \breve{\underline{\boldsymbol{f}}}_{ss} \tag{8.28}$$

Time discretization

To discretize the time derivatives, finite difference estimates are used. A semi-implicit Euler method is derived using:

$$\begin{aligned} & \frac{\boldsymbol{R}}{\Delta t^2}\left[\underline{\boldsymbol{u}}(t+\Delta t) - 2\underline{\boldsymbol{u}}(t) + \underline{\boldsymbol{u}}(t-\Delta t)\right] \\ + \quad & \frac{\boldsymbol{C}}{2\Delta t}\left[\underline{\boldsymbol{u}}(t+\Delta t) - \underline{\boldsymbol{u}}(t-\Delta t)\right] + \boldsymbol{K}\underline{\boldsymbol{u}}(t+\Delta t) = \underline{\boldsymbol{f}}(t) \end{aligned} \tag{8.29}$$

or $\hat{\boldsymbol{K}}\underline{\boldsymbol{u}}(t+\Delta t) = \hat{\underline{\boldsymbol{f}}}(t)$ where:

$$\begin{aligned} \hat{\boldsymbol{K}} &= \frac{\boldsymbol{R}}{\Delta t^2} + \frac{\boldsymbol{C}}{2\Delta t} + \boldsymbol{K} \\ \hat{\underline{\boldsymbol{f}}}(t) &= \frac{2\boldsymbol{R}}{\Delta t^2}\underline{\boldsymbol{u}}(t) + \left(\frac{\boldsymbol{C}}{2\Delta t} - \frac{\boldsymbol{R}}{\Delta t^2}\right)\underline{\boldsymbol{u}}(t-\Delta t) + \underline{\boldsymbol{f}}(t) \end{aligned} \tag{8.30}$$

The equations for a comparable explicit solution are:

$$\begin{aligned} & \frac{\boldsymbol{R}}{\Delta t^2}\left[\underline{\boldsymbol{u}}(t+\Delta t) - 2\underline{\boldsymbol{u}}(t) + \underline{\boldsymbol{u}}(t-\Delta t)\right] \\ + \quad & \frac{\boldsymbol{C}}{2\Delta t}\left[\underline{\boldsymbol{u}}(t+\Delta t) - \underline{\boldsymbol{u}}(t-\Delta t)\right] + \boldsymbol{K}\underline{\boldsymbol{u}}(t) = \underline{\boldsymbol{f}}(t) \end{aligned} \tag{8.31}$$

or

$$\begin{aligned} \underline{\boldsymbol{u}}(t+\Delta t) &= \left(\frac{\boldsymbol{R}}{\Delta t^2} + \frac{\boldsymbol{C}}{2\Delta t}\right)^{-1} \\ & \quad \left[2\frac{\boldsymbol{R}}{\Delta t^2}\underline{\boldsymbol{u}}(t) - \left(\frac{\boldsymbol{R}}{\Delta t^2} - \frac{\boldsymbol{C}}{2\Delta t}\right)\underline{\boldsymbol{u}}(t-\Delta t)\right. \\ & - \quad \left.\boldsymbol{K}\underline{\boldsymbol{u}}(t) + \underline{\boldsymbol{f}}(t)\right] \end{aligned} \tag{8.32}$$

Since both $\boldsymbol{R}$ and $\boldsymbol{C}$ are diagonal matrices, they are easily inverted.

The main feature of the explicit solution method is that the stiffness matrix $\boldsymbol{K}$ does not have to be inverted. In practice, this allows us to split the large global equation into simple independent local equations for each node:

$$\begin{aligned} \boldsymbol{u}_q(t+\Delta t) &= \left(\frac{\rho V^e}{3\Delta t^2} + \frac{\alpha\rho V^e}{6\Delta t}\right)^{-1} \\ & \quad \left[2\frac{\rho V^e}{3\Delta t^2}\boldsymbol{u}_q(t) - \left(\frac{\rho V^e}{3\Delta t^2} - \frac{\alpha\rho V^e}{6\Delta t}\right)\boldsymbol{u}_q(t-\Delta t)\right) \\ & - \quad \left\{\boldsymbol{K}\underline{\boldsymbol{u}}(t)\right\}_q + \boldsymbol{f}_q(t)\Big] \\ &= \left(\frac{1}{\Delta t} + \frac{\alpha}{2}\right)^{-1} \\ & \quad \left[\frac{2}{\Delta t}\boldsymbol{u}_q(t) - \left(\frac{1}{\Delta t} - \frac{\alpha}{2}\right)\boldsymbol{u}_q(t-\Delta t)\right) \\ & - \quad \frac{3\Delta t}{\rho V^e}\left(\left\{\boldsymbol{K}\underline{\boldsymbol{u}}(t)\right\}_q + \boldsymbol{f}_q(t)\right)\Big] \end{aligned} \tag{8.33}$$

where $\{\boldsymbol{K}\underline{\boldsymbol{u}}(t)\}_q$ is the result of the matrix-vector multiplication related to the node q.

Simulation using this equation would be similar to the mass-spring models used by eg. Kühnapfel, Deussen, and Kuhn [Deu95, Ku96, Kue93, Kue94, Kue95] and Waters and Terzopoulos [Wat87, Wat91, Ter91b, Wat92, Ter93, Lee93, Lee95].

The non-zero elements of the row associated with the q'th node of $\boldsymbol{K}$, indicate which nodes that are connected to it with springs.

The semi-implicit solution algorithm is preferred because of the better stability and more direct solution of the system. System response is more global with the implicit approach. Using explicit methods, the response to forces is only spread globally after some iterations.

A drawback with the semi-implicit models is the greater difficulty of modeling cuts. Any change in the stiffness matrix implies that the inverted stiffness matrix has to be updates or recalculated. This can be quite expensive. But with domain decomposition techniques, the domain may be split up into subdomains, which use explicit and semi-implict solution methods, depending on whether cuts have been or have not been performed in the sub-domain. See the later section 8.9.1 for a wider discussion of these aspects.

Although the discussion above has been carried out using the full system, the same equations could be formulated for the condensed system. We leave this to the reader.

8.6.2 Static system and selective matrix vector multiplication

Generally the $\underline{\hat{\boldsymbol{f}}}(t)$ vector is a full vector because of the contribution from the previous displacement vectors $\underline{\boldsymbol{u}}(t)$ and $\underline{\boldsymbol{u}}(t-\Delta t)$. In contrast the original force vector $\underline{\boldsymbol{f}}(t)$ is a sparse vector when forces only are applied to a few surface nodes. Since this is often the case in simulation, we were inspired to develop an alternative simulation method, which for sparse force vectors, is considerably faster.

The cost of this simulation method is the loss of dynamics. The idea is to use the static linear system $\boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}}$ (or the condensed version) instead of the dynamic damped harmonic motion system, and exploit the sparse structure of the static force vector.

Formally, solving the system using the inverted stiffness matrix is performed using:

$$\underline{\boldsymbol{u}} = \boldsymbol{K}^{-1}\underline{\boldsymbol{f}} \tag{8.34}$$

If only a few positions of the force vector are non-zero, clearly standard matrix vector multiplication would involve a large number of superfluous multiplications. Note that:

$$\underline{\boldsymbol{u}} = \boldsymbol{K}^{-1}\underline{\boldsymbol{f}} = \sum_i \boldsymbol{K}^{-1}_{*i}\underline{f}_i \tag{8.35}$$

where $\boldsymbol{K}^{-1}_{*i}$ is the i'th column vector of $\boldsymbol{K}^{-1}$ and f_i the i'th element of $\underline{\boldsymbol{f}}$. Since the majority of the f_i are zero, i is restricted to run through only the positions of $\underline{\boldsymbol{f}}$ for which $f_i \neq 0$. If n of the N positions in $\underline{\boldsymbol{f}}$ are non-zero this will reduce the complexity to $O(n/N)$ times the time of a normal matrix vector multiplication. We call this approach *Selective Matrix Vector Multiplication* (SMVM).

The SMVM method and the dynamic model converge for slow changes in forces applied to the solid body.

8.6.3 Summary of simulation methods

A number of methods related to simulation using FE models have been presented. In general two criterias separate the possible implicit algorithms: Full FE model contra condensed FE model, and dynamic simulation contra selective matrix vector multiplication (SMVM).

	Dynamic simulation	SMVM
Full FE model	FD	FS
Condensed FE model	CD	CS

The choice of algorithm depends on the requirements of the application. Whether one chooses dynamic simulation or SMVM is a question of the speed requirement and size of the problem. If the problem can be processed fast enough using dynamic simulation, this would be the best choice.

The choice between the full FE model versus the condensed model, is based on whether it is necessary to modify the FE model during simulation or not, eg. to model a surgical cut in the model. If modification is necessary, the standard full system is easier to modify, rather than the condensed system, which is one step further in refinement. Without the need for modification the condensed model should be used.

The CS, CD and FS methods have been implemented in the simulation system described below.

8.7 Simulation system

This section briefly discusses the practical aspects of implementing the simulation methods and generating models from voxel data. Screen shots from the SGI Performer based system are used to illustrate the results.

8.7.1 Mesh generation using Mvox and Nuages

In addition to a range of simple box-like structures, data from the Visible Human project [Vis96] has been used to make a model of a lower leg.

Since the Visible Human data set is voxel-based (see figure 8.3), it must first be turned into a mesh model. The Mvox software [BN96] is used to draw contours manually on the boundary of the skin and bone in the voxel data (see figure 8.4). The Nuages software [Gei93] is then applied to create a 3D

Figure 8.3: *Voxel data from the visible human data set.*

tetrahedral mesh model of the leg. The result for the lower leg is shown in figure 8.5. This model is used in the simulation system described in the next section.

8.7.2 SGI Performer parallel pipe-lining system

The simulation system has been implemented on a Silicon Graphics Onyx with four Mips R4400 processors using the SGI Performer 1.2 graphics library. SGI Performer allows the programmer to create parallel pipe-lining software quite easily by providing the basic tools for communication, shared memory etc.

Currently, the system runs with 3 processes (see figure 8.7): The Application, Culling and Drawing processes. The *Application* process handles the actual simulation of the deformable solid, ie. calculates displacements etc. The *Culling* process analyzes the scene that the simulation process provides, and determines which parts are visible in the current window. It then pipes the visible parts to the *Drawing* process which finally renders the scene.

Note, that although the entire system is a parallel system, the actual deformation simulation system still runs on a single processor. The parallel features are only used to separate rendering from simulation.

Figure 8.6 shows a screen dump of the virtual surgery room with the leg lying on the operating table. Figure 8.8 shows the surface of the FE mesh in the simulator.

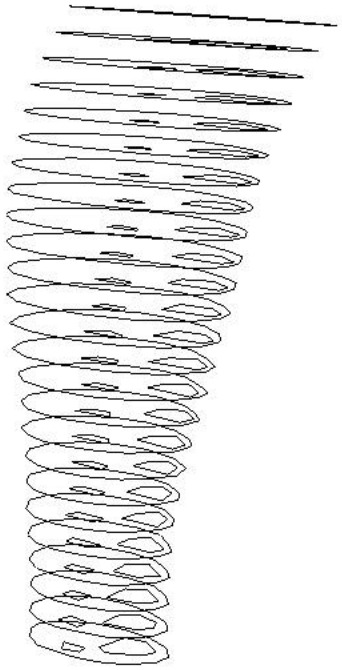

Figure 8.4: *Contours created using Mvox.*

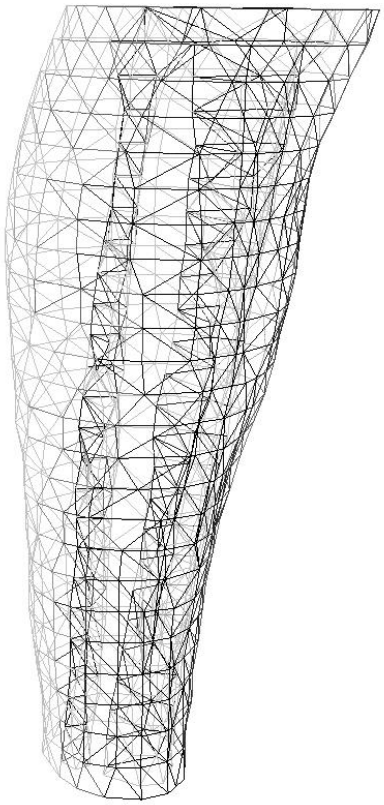

Figure 8.5: *FE mesh created from contours using Nuages.*

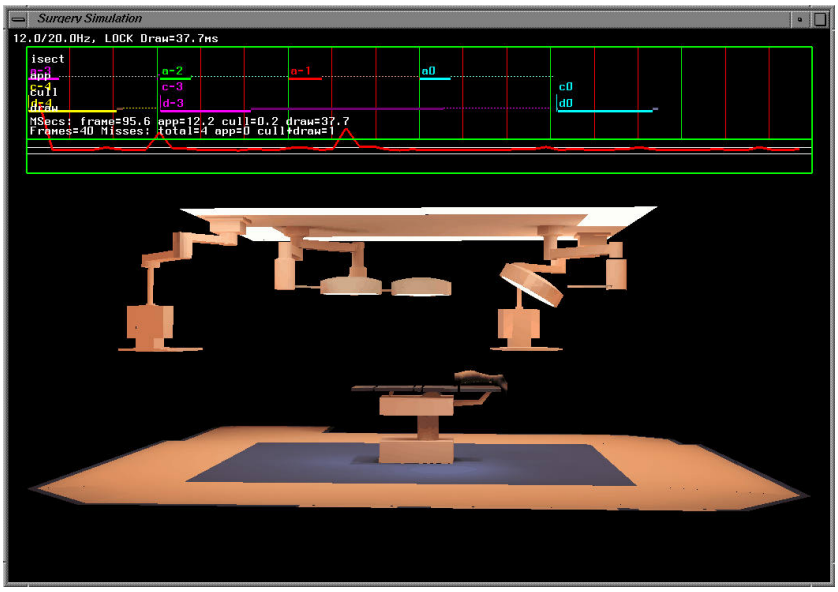

Figure 8.6: *Simulation system implemented using SGI Performer.*

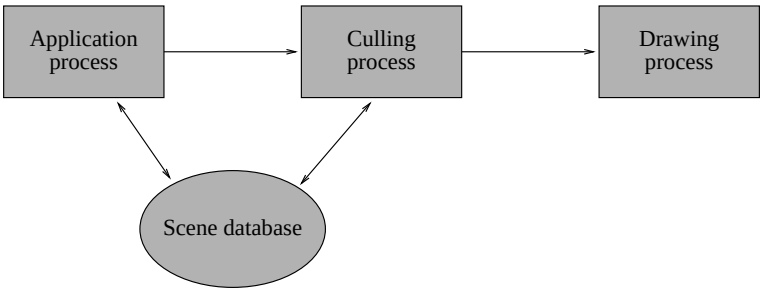

Figure 8.7: *Diagram showing relationship between the processes of the simulation system.*

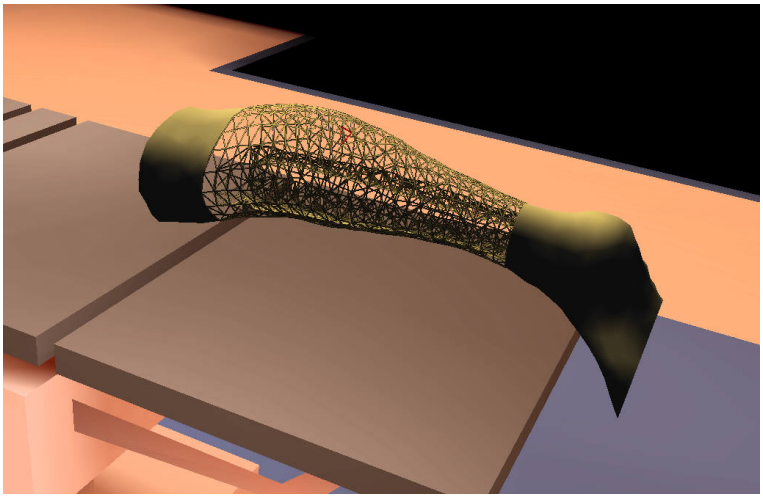

Figure 8.8: *Wireframe model of lower leg in simulator.*

8.8 Results

This chapter has described four methods for real-time simulation of elastic deformation of a volumetric solid based on semi-implicit linear elastic finite elements. Examples of deformation of a lower leg are shown in figures 8.9 and 8.10.

Performance of the dynamic simulation methods is determined solely by the size of the linear system. 20 frames/second have been achieved for models with up to 250 nodes in the system equation. For the full linear system this includes all nodes, both internally and on the surface. But for the condensed system it only includes the surface nodes. The number of internal nodes of the model, therefore, does not matter for the condensed system since they have been removed from the system equation.

It is more difficult to predict the performance of the methods using Selective Matrix Vector Multiplication (SMVM). The above comments, concerning the full contra condensed systems, apply here also. But in addition, the number of nodes which have forces applied to them is very important.

The example using a leg from the Visible Human data set with 700 system nodes (condensed system with only surface nodes) ran comfortably using only 1/3 of a frame (20 frames/second) when forces were applied to 3 nodes. This included calculation of the deformation and also basic processing. So although both more nodes and more surface nodes with forces applied, would increase the time requirement, larger models could be accommodated using the SMVM method.

As a group, we call the optimized models Fast Finite Elements (FFE).

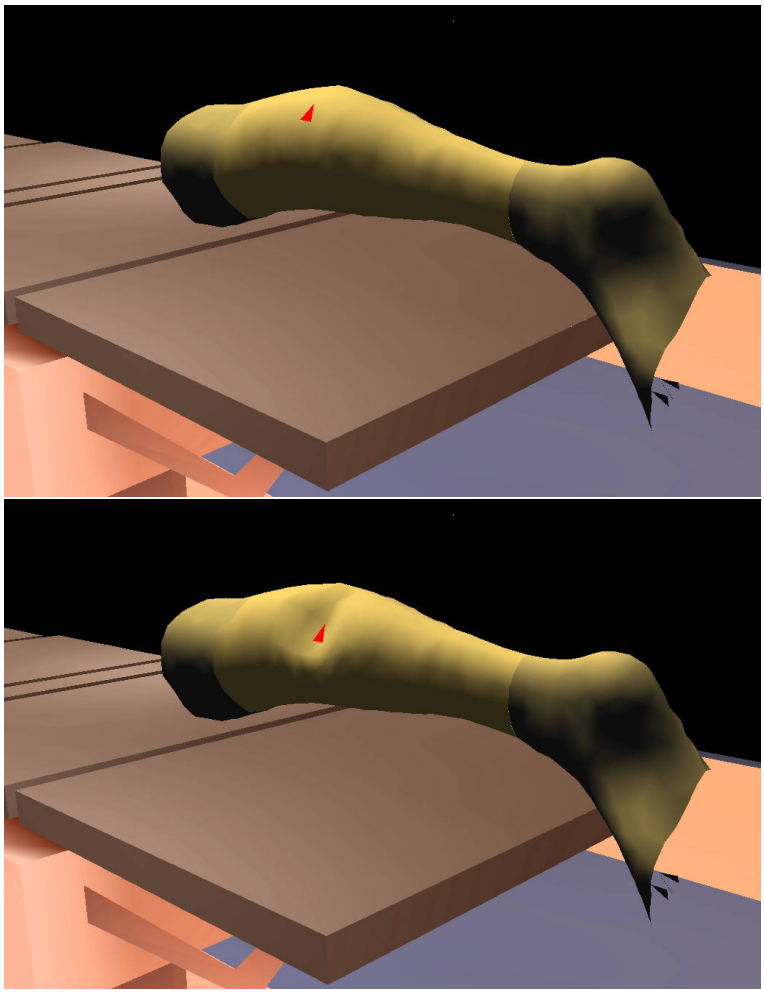

Figure 8.9: *Simulation of pushing on a the lower leg. Top: Default shape. Bottom: Deformation of leg when a push is applied to the black triangle.*

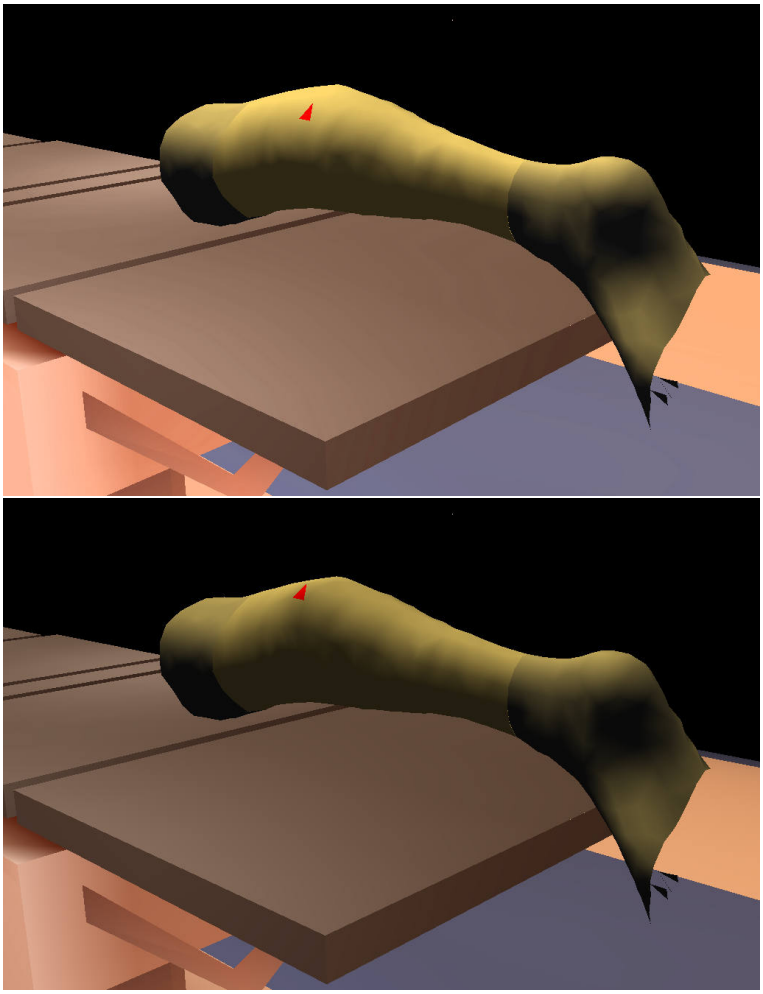

Figure 8.10: *Simulation of pulling on a the lower leg. Top: Default shape. Bottom: Deformation of leg when a push is applied to the black triangle.*

8.9 Extensions

This section describes two extensions to the basic fast finite element algorithms, which could be used in larger practical systems. Full implementation of these extensions requires considerable database handling facilities, and access to parallel systems with at least 5-10 processors.

8.9.1 Domain decomposition

The computational complexity of the finite element models is generally high. Although computational improvements have been presented in this chapter, there is still a low limit on the possible size of the deformable models, if real-time response is necessary.

Parallel processors are starting to become generally available on eg. the Silicon Graphics Onyx series of high-end graphics computers. A natural suggestion is, therefore, to create parallel versions of the finite element models.

Explicit models are very easy to parallelize, since the solution is determined locally. In principle, each node can therefore be assigned to one processor in the parallel computer.

Creating parallel versions of implicit models is more difficult. Fortunately, there is currently a great deal of attention on these problems. *Domain decomposition* techniques are being developed with the specific aim of implementing finite element models on parallel systems. Although the field is still being developed, there are reasonable methods available.

In the following, the basic ideas of domain decomposition are described. Domain decomposition methods can be quite complex, and it is not clear which method should be used for the real-time deformable models. A more detailed description is, therefore, not given here. Refer to the recent book by Smith et al. [Smi96] for a more in-depth discussion.

Let us assume that the domain Ω can be separated into non-overlapping sub-domains $\Omega^{(i)}$ with boundaries $\Gamma^{(i)}$ (see figure 8.11). Adjacent domains i and j only share the common boundary $\Gamma^{(i)} \cap \Gamma^{(j)}$.

Let the linear system for sub-domain i be:

$$\boldsymbol{K}^{(i)} \underline{\boldsymbol{u}}^{(i)} = \underline{\boldsymbol{f}}^{(i)} \tag{8.36}$$

Using the same approach as used for the condensation, the nodes of the sub-domain are split into *i*nterior and *c*ommon nodes. The common nodes are those shared by two or more sub-domains.

With these definitions, the equilibrium matrix equation for the sub-domain can be written as:

$$\begin{bmatrix} \boldsymbol{K}_{cc}^{(i)} & \boldsymbol{K}_{ci}^{(i)} \\ \boldsymbol{K}_{ic}^{(i)} & \boldsymbol{K}_{ii}^{(i)} \end{bmatrix} \begin{bmatrix} \underline{\boldsymbol{u}}_c^{(i)} \\ \underline{\boldsymbol{u}}_i^{(i)} \end{bmatrix} = \begin{bmatrix} \underline{\boldsymbol{f}}_c^{(i)} \\ \underline{\boldsymbol{f}}_i^{(i)} \end{bmatrix} \tag{8.37}$$

From this block matrix system a new linear matrix system may be derived,

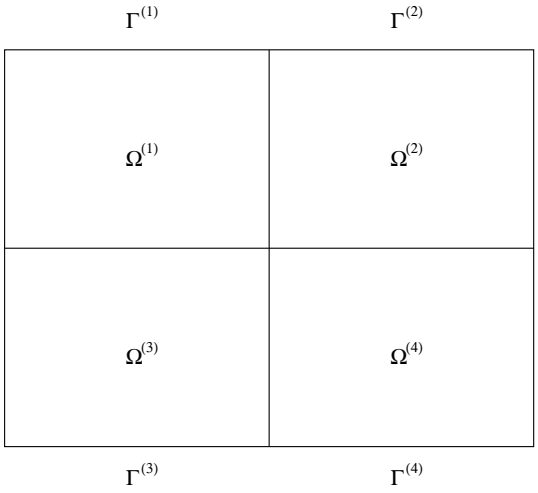

Figure 8.11: *Domain decomposition.*

which only involves the variables of the common nodes:

$$\breve{\boldsymbol{K}}_{cc}^{(i)} \underline{\boldsymbol{u}}_c^{(i)} = \breve{\underline{\boldsymbol{f}}}_c^{(i)} \tag{8.38}$$

where

$$\breve{\boldsymbol{K}}_{cc}^{(i)} = \boldsymbol{K}_{cc}^{(i)} - \boldsymbol{K}_{ci}^{(i)} \boldsymbol{K}_{ii}^{(i)-1} \boldsymbol{K}_{ic}^{(i)} \tag{8.39}$$

$$\breve{\underline{\boldsymbol{f}}}_c^{(i)} = \underline{\boldsymbol{f}}_c^{(i)} - \boldsymbol{K}_{ci}^{(i)} \boldsymbol{K}_{ii}^{(i)-1} \underline{\boldsymbol{f}}_i^{(i)} \tag{8.40}$$

The sub-domain variables are assembled into a global system

$$\breve{\boldsymbol{K}}_{cc} \boldsymbol{u}_c = \breve{\underline{\boldsymbol{f}}}_c \tag{8.41}$$

where $\breve{\boldsymbol{K}}_{cc}$ is called the *Schur's complement.*

Finally, when a solution has been found for the common nodes, the displacements of the internal nodes in the sub-domain can be calculated using:

$$\underline{\boldsymbol{u}}_i^{(i)} = \boldsymbol{K}_{ii}^{(i)-1} (\underline{\boldsymbol{f}}_i^{(i)} - \boldsymbol{K}_{ic}^{(i)} \underline{\boldsymbol{u}}_c^{(i)}) \tag{8.42}$$

In theory, the following approach could be used to solve the global matrix equation using domain decomposition (*Parallel* indicates that the step can be carried out in parallel):

1. *Parallel*: The common forces are determined locally in each sub-domain using equation 8.40,

2. These forces are assembled into a global force vector, which is used to find the displacements of the common nodes using equation 8.41,

3. *Parallel*: When the displacements of the common nodes are found, the displacements of the internal nodes can be determined using equation 8.42.

Normal domain decomposition methods do not use this approach directly. Instead iterated conjugate gradient methods [Brt95] are often used with preconditioners based on the Schur's complement.

The actual application will determine what form of domain decomposition algorithm should be used. In the case of cutting, the Schur's complements are changed and iterated methods are probably necessary.

Experiments have been carried out to solve the global linear system:

$$\boldsymbol{K}\underline{\boldsymbol{u}} = \underline{\boldsymbol{f}} \tag{8.43}$$

by applying domain decomposition based on the conjugate gradient algorithm.

First a preconditioner is created from the global stiffness matrix by zeroing the off-diagonal elements, which are related to the common nodes. Since the sub-domain blocks of the preconditioner are now independent, the application of the preconditioner in the conjugate gradient algorithm, can be carried out in parallel by sub-domain processors. The remaining part of the conjugate gradient algorithm is global, and can be performed on a specially assigned processor.

Preliminary results have been obtained showing the validity of the algorithm. But, the conjugate gradient algorithm is slow, and the solution of the global system was not fast enough for real-time performance. More work is therefore necessary, to determine an appropriate domain decomposition algorithm.

8.9.2 Cutting in finite element systems

Implementing cutting in finite element systems causes two major problems. The first, which we ignore here, is related to the geometric modification of the finite element mesh. These modifications should ensure that the cut looks smooth when rendered using computer graphics. In general, it is necessary to refine the mesh around the cut, and the geometric aspects of this are non-trivial.

The second problem concerns the necessary change of the stiffness matrix and the linear system. For explicit methods this is simple. But, for implicit systems, where the stiffness matrix has been inverted, the inverted stiffness matrix needs to be updated.

For simplicity, let us assume that the cutting implies removing the finite element number e from the system. The modified stiffness matrix becomes:

$$\boldsymbol{K} = \boldsymbol{K} - \text{global}(\boldsymbol{K}^e) \tag{8.44}$$

Since the size of the element stiffness matrix is 12×12, this involves changes in 12 rows and 12 columns.

To change the inverted stiffness matrix, the Woodbury formula [Num92] is used. The Woodbury formula is a matrix extension of the original scalar Sherman-Morrison update formula.

Let us assume that the stiffness matrix is updated with a matrix, which can be written as the outer product of two $3N \times 12$ vectors (N is the number of nodes):

$$\boldsymbol{K} \rightarrow \boldsymbol{K} + \boldsymbol{U}\boldsymbol{V}^T \tag{8.45}$$

The updated inverted stiffness matrix then becomes:

$$\begin{aligned} (\boldsymbol{K} + \boldsymbol{U}\boldsymbol{V}^T)^{-1} &= \boldsymbol{K}^{-1} && (8.46) \\ &- \left[\boldsymbol{K}^{-1}\boldsymbol{U}(\boldsymbol{I}_{12\times 12} + \boldsymbol{V}^T\boldsymbol{K}^{-1}\boldsymbol{U})^{-1}\boldsymbol{V}^T\boldsymbol{K}^{-1}\right] && (8.47) \end{aligned}$$

where $\boldsymbol{I}_{12\times 12}$ is a 12 by 12 identity matrix.

The vectors $\boldsymbol{U}$ and $\boldsymbol{V}$ are easily determined: $\boldsymbol{K}^e$ and the identity matrix are inserted in the rows of $\boldsymbol{U}$ and $\boldsymbol{V}$, respectively, corresponding to the global row/column positions of the element nodes.

The computation consists of the inversion of the 12 by 12 matrix:

$$\boldsymbol{I}_{12\times 12} + \boldsymbol{V}^T\boldsymbol{K}^{-1}\boldsymbol{U} \tag{8.48}$$

and 6 matrix multiplications of different sizes and two matrix additions. In total $324N^2 + O(N)$ flops.

To demonstrate the applicability of this approach, a simple cutting algorithm has been implemented in the simulation system. This algorithm allows individual finite elements to be removed. The resulting modification of the linear system is obtained using the algorithm described above.

An example is shown in figure 8.12. This simple finite element model has only 75 nodes and the modification of the linear system is more or less instant. For a more complex mesh with 1125 nodes, the modification of the inverted stiffness matrix takes about one minute. The time consumption is, consequently, too high for general real-time response. But, it could be used in specific cases, where the waiting time can be accepted or otherwise disguised.

8.10 Summary

In this chapter, a comprehensive description of the issues facing a developer of real-time deformable models has been given. In particular, new results using condensation techniques, direct matrix inversion, and selective matrix vector multiplication were presented. These methods are essential to obtaining acceptable response in surgery simulation systems.

A simulation system developed using SGI Performer was described, and a routine for creating the basic finite element mesh was suggested.

Two important extensions of the work were outlined in the final section. These extensions are the use of domain decomposition for parallel implementation of the finite element system, and the procedure for modification of the system matrix in response to cuts in the mesh.

A simple example of a cut were demonstrated, in which a single finite element was removed from the stiffness matrix and a hole appeared.

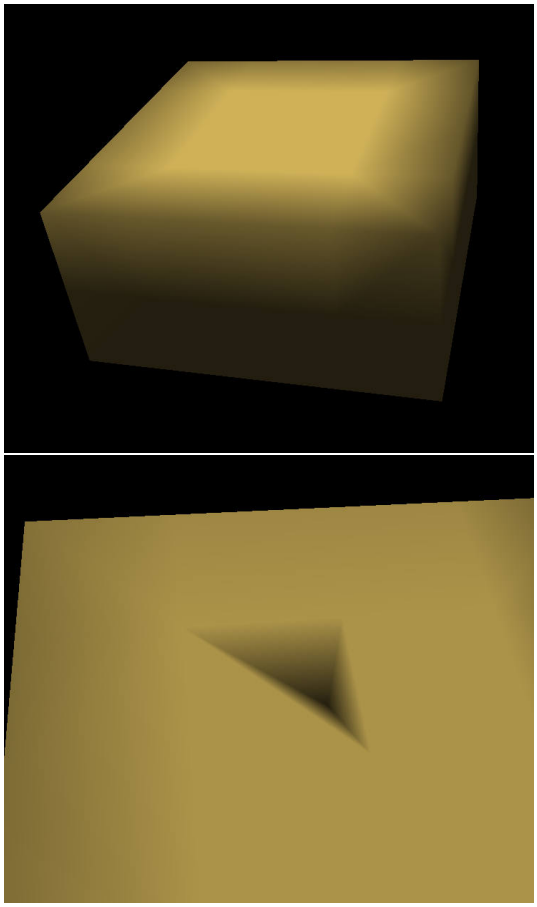

Figure 8.12: *Simple cutting in finite element mesh.*

The implementations of the two extensions are non-trivial and require comprehensive book-keeping and computer power to work in practice. Both extensions, consequently, need more work before they can be used in practice.

In general, there is still a considerable amount of work to be performed. An important example is the introduction of detailed segmentations of the organs, limbs etc. to allow different material properties and models to be used. The current parallel research in digital atlasses such as the VoxelMan atlas [Hoe92] is a significant step in this direction.

Chapter 9

Conclusion

In this thesis, medical image registration and surgery simulation techniques have been explored. Reviews of previous work in these fields were given in chapters 2 and 7.

If rigid transformation is understood as physical motion, physical models has been the common denominator of all the work presented here.

In general, physical models are becoming more and more important in the fields of image processing, computer graphics, medical imaging, and simulation. This is happening, because in all these fields there is a movement in these years, from regarding the world as consisting of rigid objects, to taking the natural path of regarding the world as simply - the world.

The environment is, therefore, increasingly being simulated and interpretated using non-rigid modeling techniques, which have their basis in physical models of natural phenomena.

Often the physical models have been used without regard to their actual applicability for the specific problem. This is natural when new techniques are introduced to a field, but unfortunate in the long run, since problems inevitably occur. This text has, therefore, been written with a strong emphasis on *really* understanding the physical models and their limitations. In particular, chapter 4 were dedicated to a in-depth description of physical continuum models.

Some of the results in this thesis have come about *because* this understanding opened up new ways of handling the specific problems. This was seen in chapter 5, where the viscous fluid registration algorithm were improved using results from linear elasticity. This transfer of knowledge from one model to another, was made possible by a general understanding of continuum models.

In addition, the implementation of the linear elasticity operator as a filter, depended on an understanding of the characteristics of quadratic energy functions, and their linear partial differential equations.

Sometimes, it is not possible to use the appropriate physical model because of computational complexity considerations. The real-time finite element models of linear elasticity in chapter 8 are a good example of a case where a model is selected entirely based on timing concerns. But, although the specific appli-

cation without doubt violates basic assumptions of the model, the knowledge of the continuum models allows us to *understand* when and how these assumptions are violated. Thus, we are able to make an intelligent trade-off, rather than a blind decision.

In some cases, the real physical model is not known. Naturally, this hampers the use of realistic models, and often an arbitrary model is used instead. This has typicly happened for non-rigid registration of medical images. Since no model of the change from one image to the next is generally available, elastic and viscous fluid models have been used instead.

But, in isolated cases it is possible to determine the real physical model. Chapter 6 described such a case, where theories about the real model were used to build a specific model for the problem at hand. In this chapter, time sequence images of the mandibular bone were registered using a growth model, based on medical knowledge of the growth processes of bone.

Finally, there are alternative aspects to this thesis than the physical models. Physical models have been the theoretical hinge of thesis. But, the thesis can also be seen from an application point of view. Indeed, as argued in the introduction, the image registration and simulation algorithms presented here, can be seen as some of the basic elements of the next generation medical workstation. A workstation, where all the components of visualization, automated diagnosis, surgery simulation, and surgery assistance, come together.

Whether this workstation will ever see the light is another question. There is still a long way to go, and this thesis has only solved a small part of the problems. Much of the next generation workstation is still visions of the future of medicine.

There is a lot of work in front of us.

Appendix A

Mvox

Mvox is a general-purpose tool for visualization, segmentation and manipulation of a wide range of 2-4D greylevel and color images, and 3D surface graphics. The software provides a flexible tool, that handles all the data types typically used in a research environment, for medical imaging and visualization.

Mvox is easy to use, and has a consistent user interface, based on software standards such as Unix, X/Motif and OpenGL.

Mvox can handle images with different numbers of slices, colors, and time steps. It has been successfully used to visualize and segment:

- 2D and 3D medical images with stacks of slices (CT, MRI, etc).
- 2D and 3D color images, such as histological slices from the Visible Human data set.
- 2D remote sensing images with many channels,
- 3D surface graphics, such as those used in CAD software and produced by the iso-surface routine in Mvox [Wyv86] (similar to Marching Cubes [Lor87]).
- time sequences of any of the above.

In Mvox, image segmentation can be performed manually using editing and contour drawing facilities (see figure A.3). For semi-automatic segmentation, Mvox has interactive/automatic thresholding (see figure A.2) and statistical classification using discriminant analysis [Con84], both with multiple classes.

Segmented images can be visualized in 3D, along with the original images, using either iso-surfacing [Wyv86], volume rendering or fast display of slices (see figure A.4 and A.5). Also stacked contours can be turned into 3D structures.

Additional features include image registration, computation and manipulation of image histograms, computation of image statistics, colormap alteration, etc. User-developed C programs can use Mvox as a front-end through a shared memory interface by including as little as two lines of extra user code. All this makes Mvox a very flexible tool suited to a wide variety of user needs.

Images in this appendix show different screen-shots from Mvox.

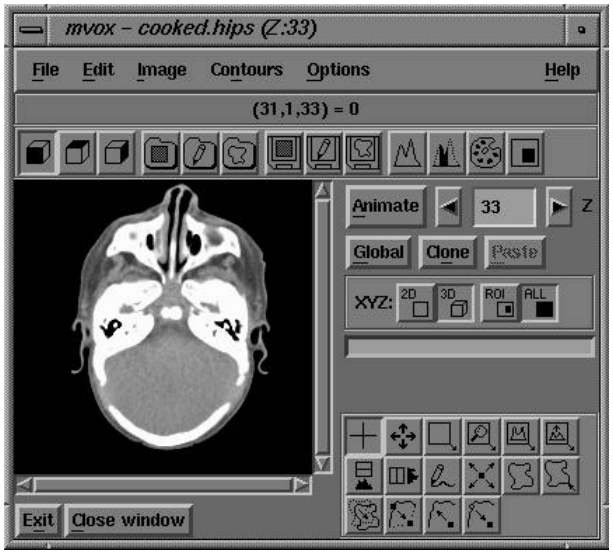

Figure A.1: *Mvox main 2D image handling window.*

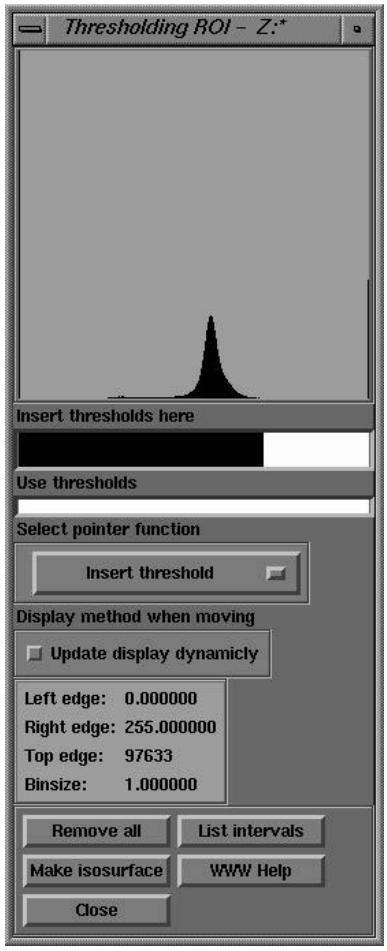

Figure A.2: *Mvox thresholding window.*

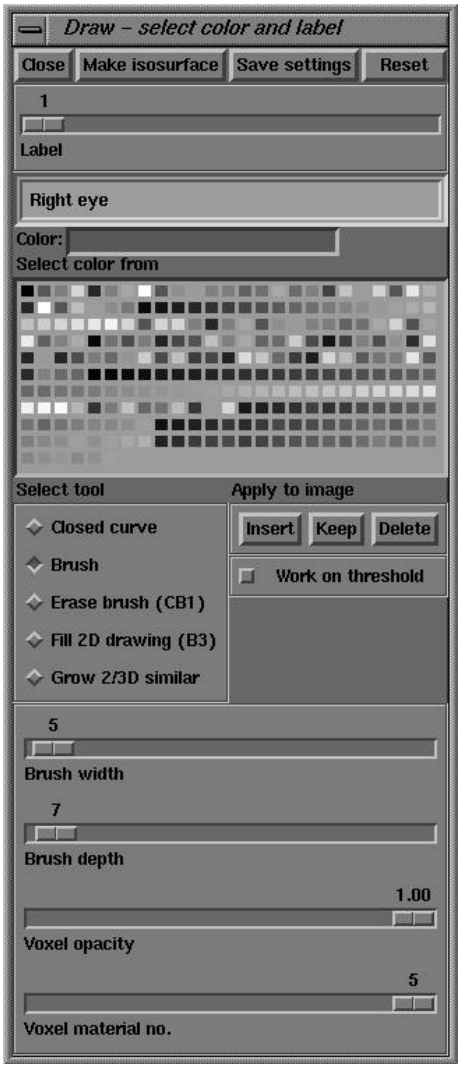

Figure A.3: *Mvox drawing window.*

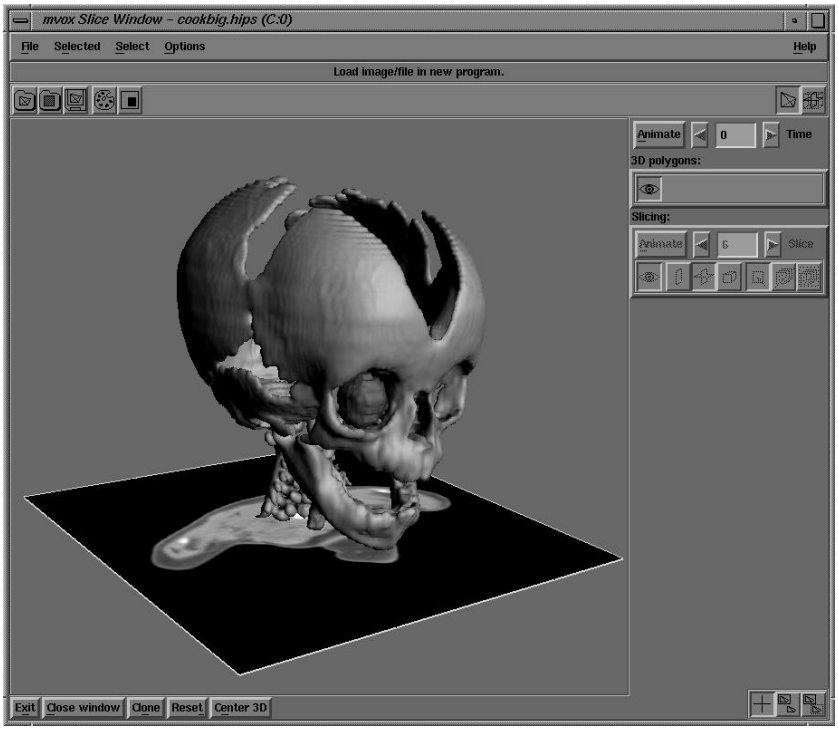

Figure A.4: *Mvox 3D graphics window.*

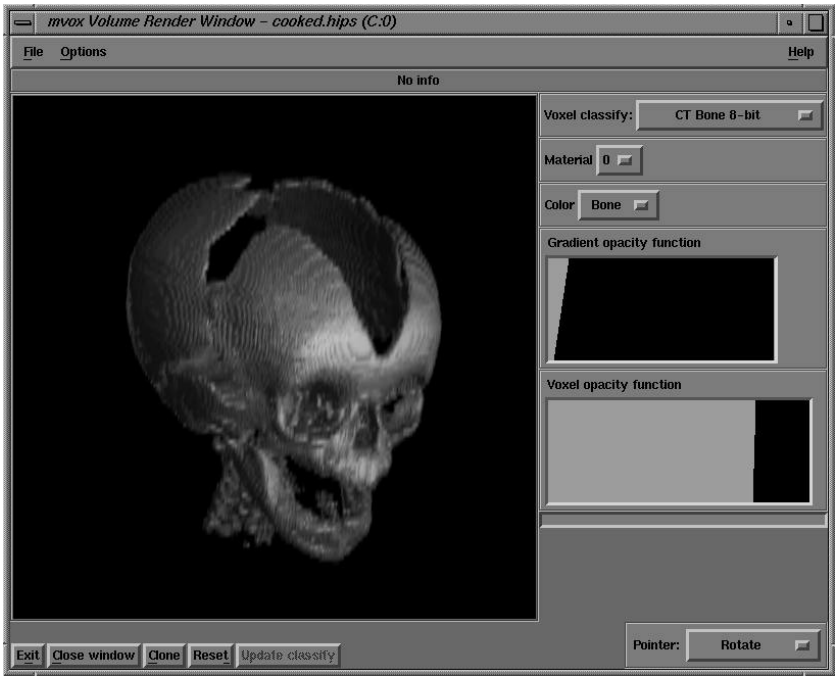

Figure A.5: *Mvox 3D volume rendering window.*

Appendix B

Derivation of Linear Filter.

The decomposition of the impulse response displacement field is determined by inserting:

$$\begin{aligned} \nu_{lmns} &= -\frac{1}{\kappa_{lmns}} < \tilde{\boldsymbol{f}}_c, \boldsymbol{\phi}_{lmns}(\tilde{\boldsymbol{x}}) > \\ &= -\frac{1}{\kappa_{lmns}} \phi^{x_1}_{lmns}(\tilde{\boldsymbol{x}}_c) \end{aligned} \tag{B.1}$$

into the decomposition equation.

$$\tilde{\boldsymbol{v}}(\tilde{\boldsymbol{x}}) = \sum_{ijk}^{\infty} \sum_{r=1}^{3} \nu_{ijkr} \boldsymbol{\phi}_{ijkr}(\tilde{\boldsymbol{x}}) \tag{B.2}$$

$$= \sum_{ijk}^{\infty} \sum_{r=1}^{3} -\frac{1}{\kappa_{ijkr}} \phi^{x_1}_{ijkr}(\tilde{\boldsymbol{x}}_c) \boldsymbol{\phi}_{ijkr}(\tilde{\boldsymbol{x}}) \tag{B.3}$$

$$\begin{aligned} = \frac{1}{\pi^2 \mu(2\mu+\lambda)} \sum_{ijk}^{\infty} \frac{\mathrm{scc}(\tilde{\boldsymbol{x}})}{i^2+j^2+k^2} \\ \left(\mu\alpha_1^2 \begin{bmatrix} i^2\mathrm{scc}(\tilde{\boldsymbol{x}}) \\ ij\mathrm{csc}(\tilde{\boldsymbol{x}}) \\ ik\mathrm{ccs}(\tilde{\boldsymbol{x}}) \end{bmatrix} \right. \\ + (2\mu+\lambda)\alpha_2^2 \begin{bmatrix} j^2\mathrm{scc}(\tilde{\boldsymbol{x}}) \\ -ij\mathrm{csc}(\tilde{\boldsymbol{x}}) \\ 0 \end{bmatrix} \\ \left. + (2\mu+\lambda)\alpha_3^2 \begin{bmatrix} (ik)^2\mathrm{scc}(\tilde{\boldsymbol{x}}) \\ ijk^2\mathrm{csc}(\tilde{\boldsymbol{x}}) \\ -ik(i^2+j^2)\mathrm{ccs}(\tilde{\boldsymbol{x}}) \end{bmatrix} \right) \end{aligned} \tag{B.4}$$

$$= \frac{8}{\pi^2 \mu(2\mu+\lambda)} \sum_{ijk}^{\infty} \frac{\mathrm{scc}(\tilde{\boldsymbol{x}})}{(i^2+j^2+k^2)^2 \Gamma_{ijk}}$$

$$\begin{bmatrix} \left(\mu i^2 + \frac{2\mu+\lambda}{i^2+j^2}((i^2+j^2+k^2)j^2 + (ik)^2)\right) \mathrm{scc}(\tilde{\boldsymbol{x}}) \\ \left(\mu ij + \frac{2\mu+\lambda}{i^2+j^2}(-(i^2+j^2+k^2)ij + ijk^2)\right) \mathrm{csc}(\tilde{\boldsymbol{x}}) \\ \left(\mu ik - \frac{2\mu+\lambda}{i^2+j^2} ik(i^2+j^2)\right) \mathrm{ccs}(\tilde{\boldsymbol{x}}) \end{bmatrix} \tag{B.5}$$

$$= \frac{8}{\pi^2\mu(2\mu+\lambda)} \sum_{ijk}^{\infty} \frac{\mathrm{scc}(\tilde{\boldsymbol{x}}_c)}{(i^2+j^2+k^2)^2\Gamma_{ijk}}$$

$$\begin{bmatrix} (\mu i^2 + (2\mu+\lambda)(j^2+k^2))\mathrm{scc}(\tilde{\boldsymbol{x}}) \\ -(\mu+\lambda)ij\mathrm{csc}(\tilde{\boldsymbol{x}}) \\ -(\mu+\lambda)ik\mathrm{ccs}(\tilde{\boldsymbol{x}}) \end{bmatrix} \tag{B.6}$$

Appendix C

Linear Tetrahedral Finite Element

We assume that the nodes of the tetrahedron have been numbered as illustrated in figure 8.2. The *natural coordinates* L_1, L_2, L_3 and L_4 of the tetrahedron are related to the global coordinates x, y and z by (we ignore element superscripts):

$$\begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{bmatrix} \begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} \tag{C.1}$$

This equation can be inverted to give

$$\begin{bmatrix} L_1 \\ L_2 \\ L_3 \\ L_4 \end{bmatrix} = \frac{1}{6V} \begin{bmatrix} a_1 & b_1 & c_1 & d_1 \\ a_2 & b_2 & c_2 & d_2 \\ a_3 & b_3 & c_3 & d_3 \\ a_4 & b_4 & c_4 & d_4 \end{bmatrix} \begin{bmatrix} 1 \\ x \\ y \\ z \end{bmatrix} \tag{C.2}$$

where

$$\begin{aligned} 6V &= \begin{vmatrix} 1 & 1 & 1 & 1 \\ x_1 & x_2 & x_3 & x_4 \\ y_1 & y_2 & y_3 & y_4 \\ z_1 & z_2 & z_3 & z_4 \end{vmatrix} \qquad a_1 = \begin{vmatrix} x_2 & x_3 & x_4 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{vmatrix} \\ b_1 &= -\begin{vmatrix} 1 & 1 & 1 \\ y_2 & y_3 & y_4 \\ z_2 & z_3 & z_4 \end{vmatrix} \qquad c_1 = \begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_4 \\ z_2 & z_3 & z_4 \end{vmatrix} \\ d_1 &= -\begin{vmatrix} 1 & 1 & 1 \\ x_2 & x_3 & x_4 \\ y_2 & y_3 & y_4 \end{vmatrix} \end{aligned} \tag{C.3}$$

The other coefficients are found by cyclic interchange of the indices.

The $\boldsymbol{B}^e$ matrix becomes:

$$\boldsymbol{B}^e = \frac{1}{6V}\begin{bmatrix} b_1 & 0 & 0 & b_2 & 0 & 0 & b_3 & 0 & 0 & b_4 & 0 & 0 \\ 0 & c_1 & 0 & 0 & c_2 & 0 & 0 & c_3 & 0 & 0 & c_4 & 0 \\ 0 & 0 & d_1 & 0 & 0 & d_2 & 0 & 0 & d_3 & 0 & 0 & d_4 \\ c_1 & b_1 & 0 & c_2 & b_2 & 0 & c_3 & b_3 & 0 & c_4 & b_4 & 0 \\ 0 & d_1 & c_1 & 0 & d_2 & c_2 & 0 & d_3 & c_3 & 0 & d_4 & c_4 \\ d_1 & 0 & b_1 & d_2 & 0 & b_2 & d_3 & 0 & b_3 & d_4 & 0 & b_4 \end{bmatrix} \tag{C.4}$$

Appendix D

GLCM Plots

D.1 MR-Pd / MR-T1 (moved)

Table D.1: *Pd/T1: Regression slope of rotation and translation experiments. Calculated correction factor. 500 samples are used.*

	Rotation	Translation	Correction
Energy	0.004092872	0.002872422	1.4249
Variance	0.00409286	0.002872399	1.4249
Entropy	0.03425994	0.02847779	1.2030
MI	0.01463079	0.01619386	0.9035
IDM	0.004175296	0.00336414	1.2411
Inertia	22.96762	28.85428	0.7960
Dmoment	20.0902	21.56672	0.9315
Correlation	0.01217191	0.01629539	0.7470
Cshade	4154.166	5863.871	0.7084
Cprominence	1120958	1408187	0.7960
Woods MR/PET (X:Pd)	0.1528903	0.1322085	1.1564
Woods MR/PET (X:T1)	0.1832248	0.1417444	1.2926

Table D.2: *Pd/T1: Corrected linear regression results ($y = ax+b$). 500 samples are used.*

	Slope (a)	Intercept (b)
Energy	0.002860538	-0.310261424
Variance	0.002860496	-0.310235623
Entropy	0.02886965	2.86163008
MI	0.01669293	-0.69676310
IDM	0.003402034	-0.770857566
Inertia	29.40917	134.97652
Dmoment	21.98069	516.35668
Correlation	0.01652248	-0.94779800
Cshade	5985.225	-414008.242
Cprominence	1434930	-71745585
Woods MR/PET (X:Pd)	0.1300614	0.9164917
Woods MR/PET (X:T1)	0.1390098	1.6041555

Table D.3: *Pd/T1: Similarity measure plot quality result and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	2	0.9384729	0.9088758
Variance	2	0.9384698	0.9088723
Entropy	5	0.9760593	0.9708134
MI	5	0.9260448	0.9214459
IDM	5	0.9685928	0.9599456
Inertia	4	0.9173130	0.9061781
Dmoment	4	0.9187858	0.9159433
Correlation	4	0.9057705	0.8883932
Cshade	3	0.8380936	0.8114711
Cprominence	3	0.8265533	0.8120783
Woods MR/PET (X:Pd)	3	0.7361181	0.7412743
Woods MR/PET (X:T1)	3	0.5480094	0.5482602

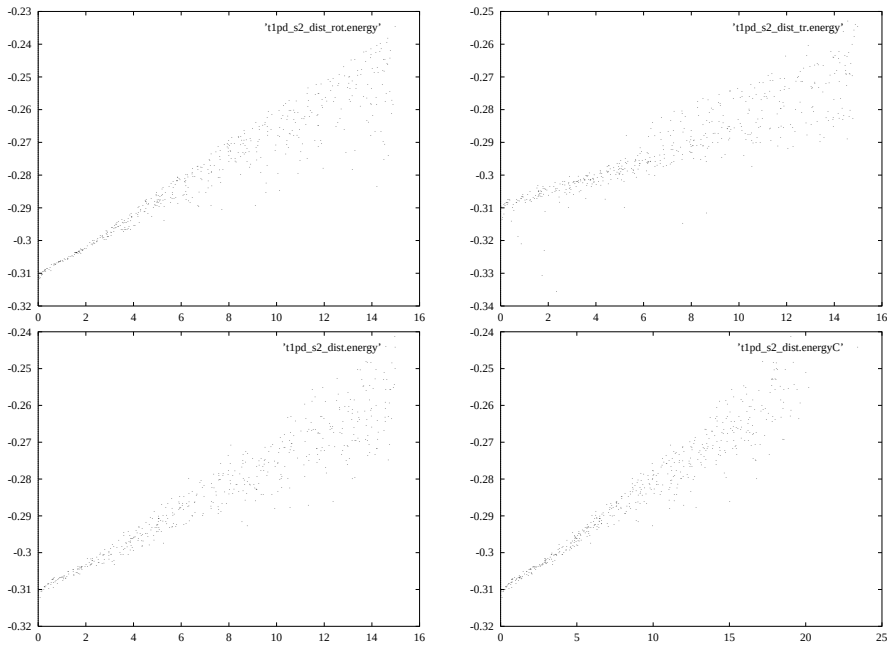

Figure D.1: *Pd/T1: Distance/energy: Rotation only, Translation only, Normal and Corrected.*

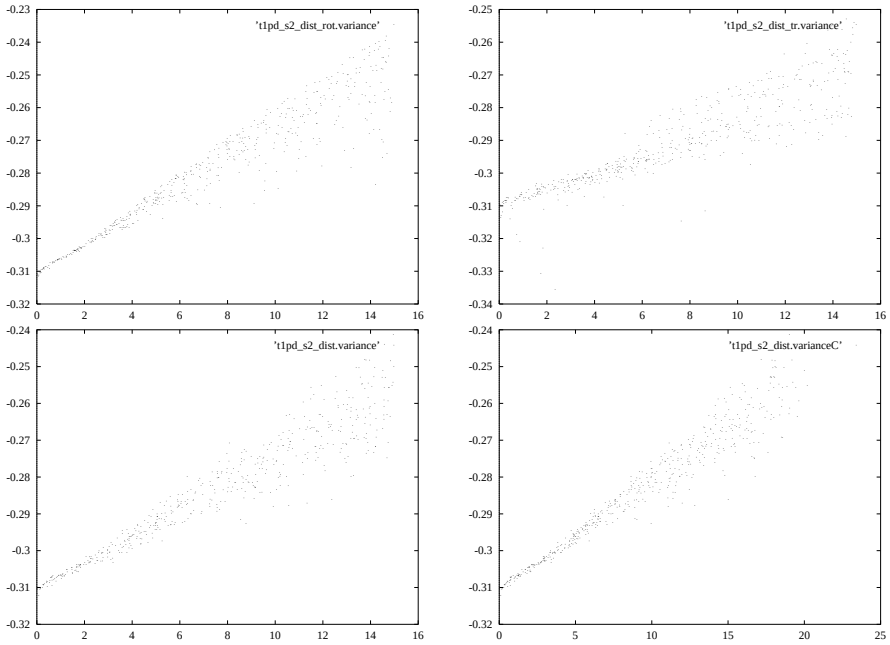

Figure D.2: *Pd/T1: Distance/variance: Rotation only, Translation only, Normal and Corrected.*

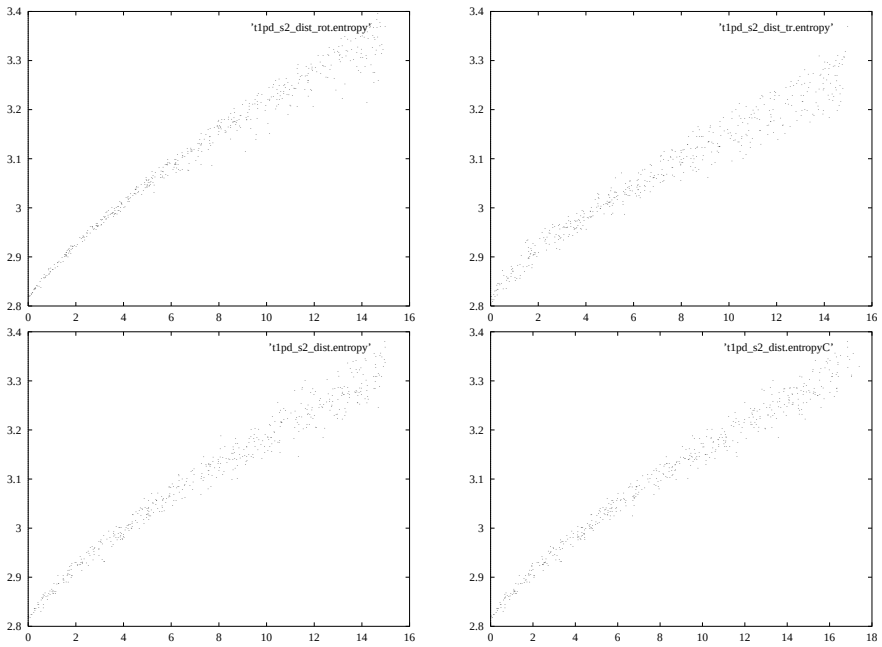

Figure D.3: *Pd/T1: Distance/entropy: Rotation only, Translation only, Normal and Corrected.*

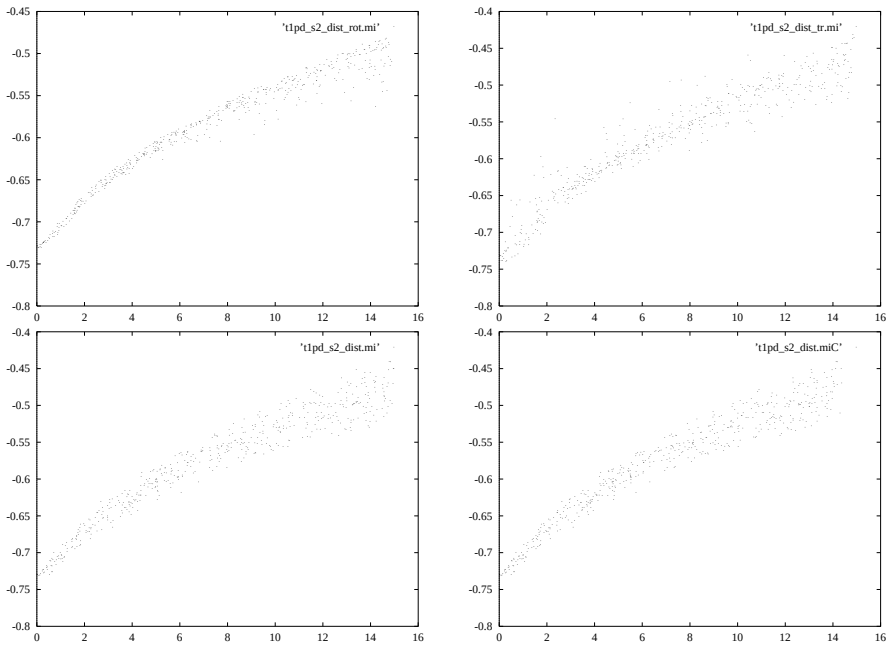

Figure D.4: *Pd/T1: Distance/MI: Rotation only, Translation only, Normal and Corrected.*

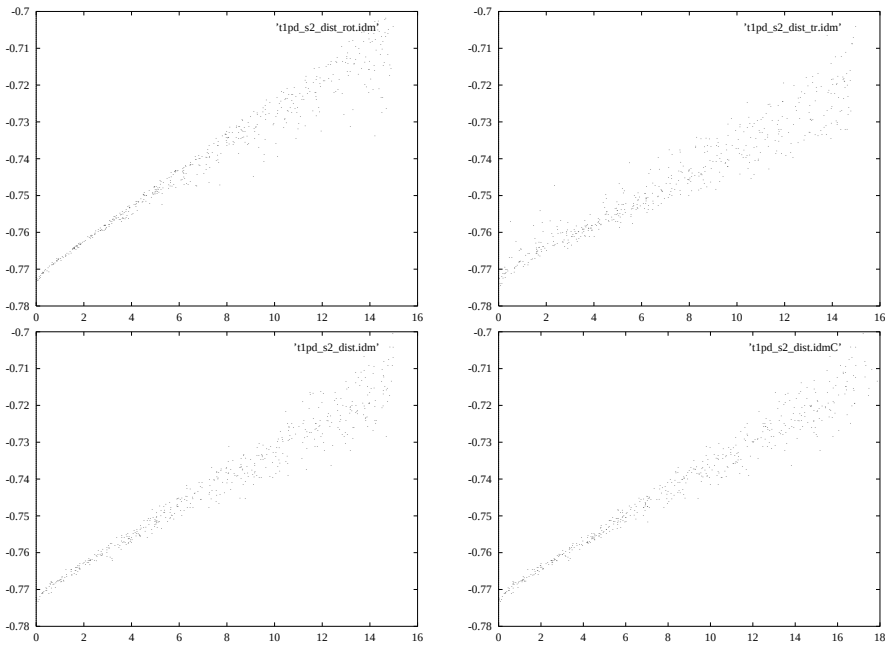

Figure D.5: *Pd/T1: Distance/IDM: Rotation only, Translation only, Normal and Corrected.*

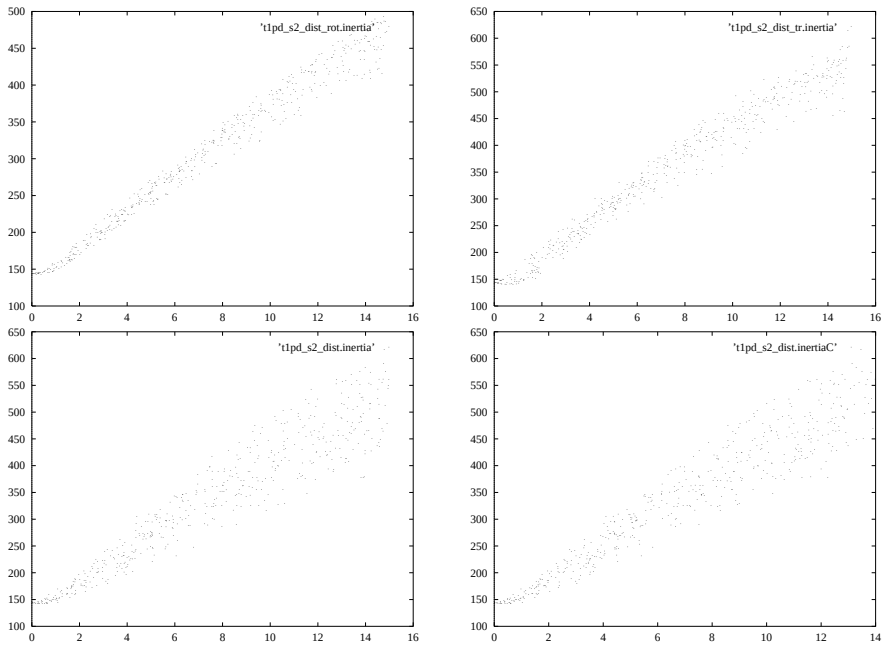

Figure D.6: *Pd/T1: Distance/inertia: Rotation only, Translation only, Normal and Corrected.*

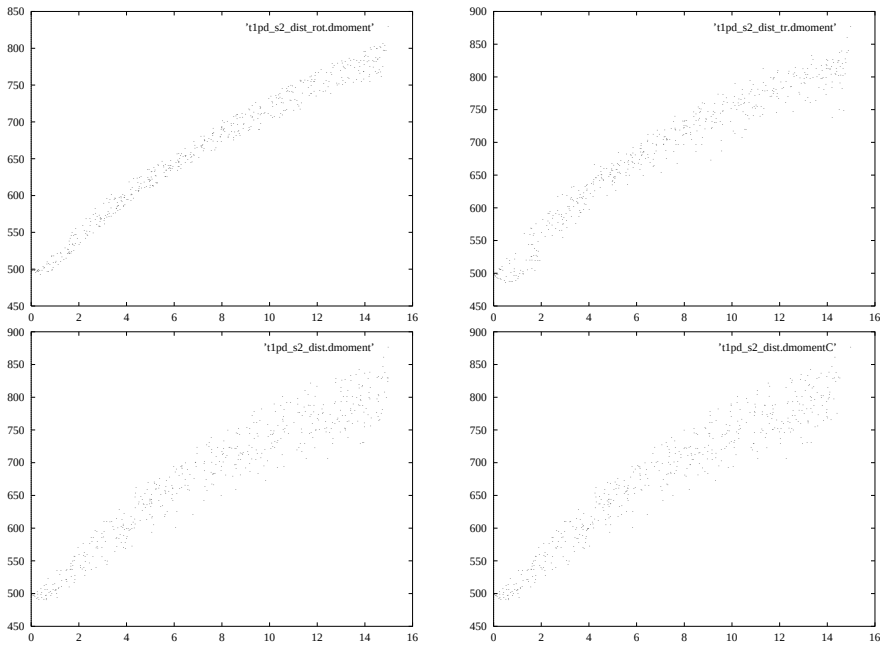

Figure D.7: *Pd/T1: Distance/Dmoment: Rotation only, Translation only, Normal and Corrected.*

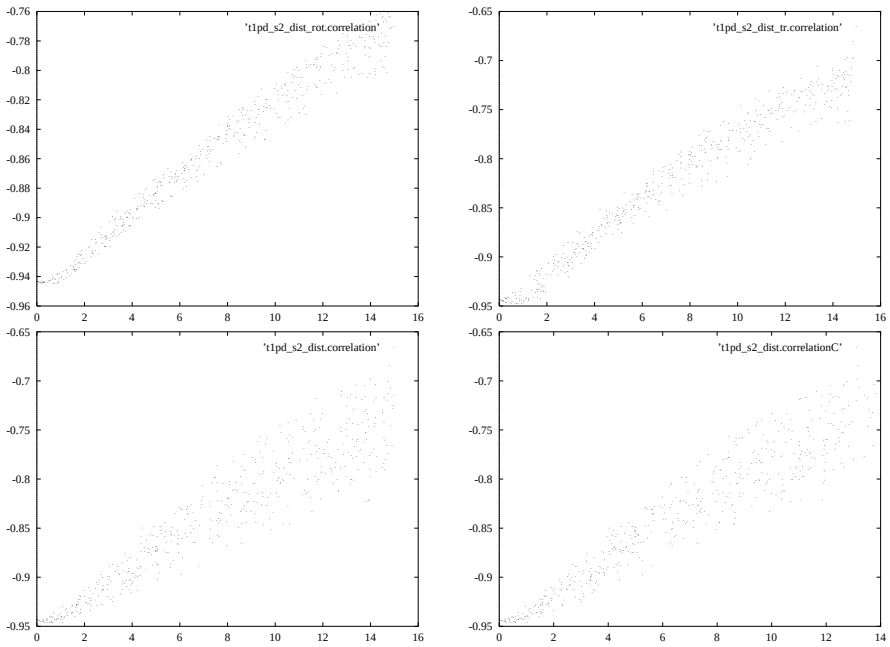

Figure D.8: *Pd/T1: Distance/correlation: Rotation only, Translation only, Normal and Corrected.*

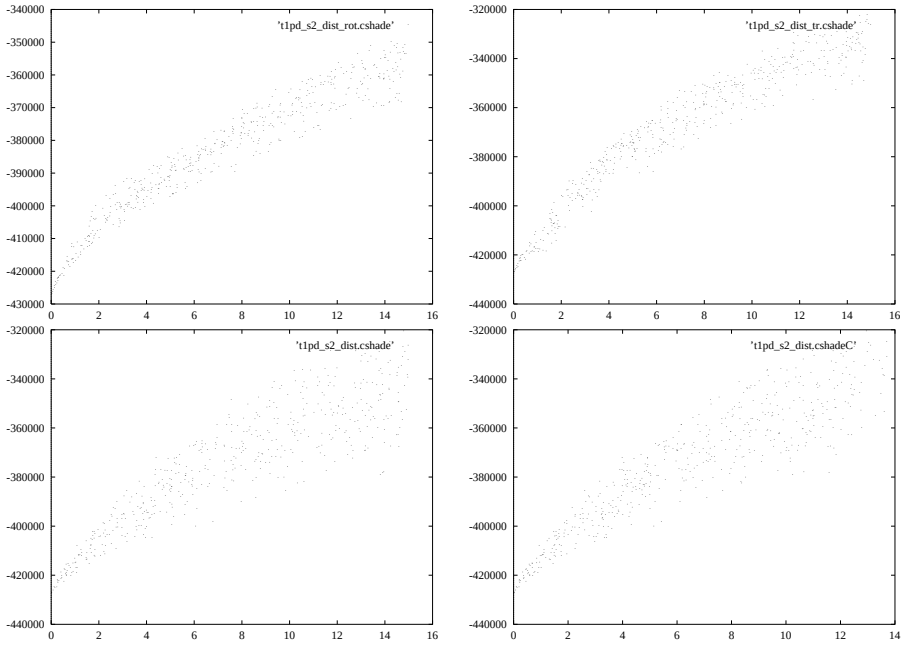

Figure D.9: *Pd/T1: Distance/Cshade: Rotation only, Translation only, Normal and Corrected.*

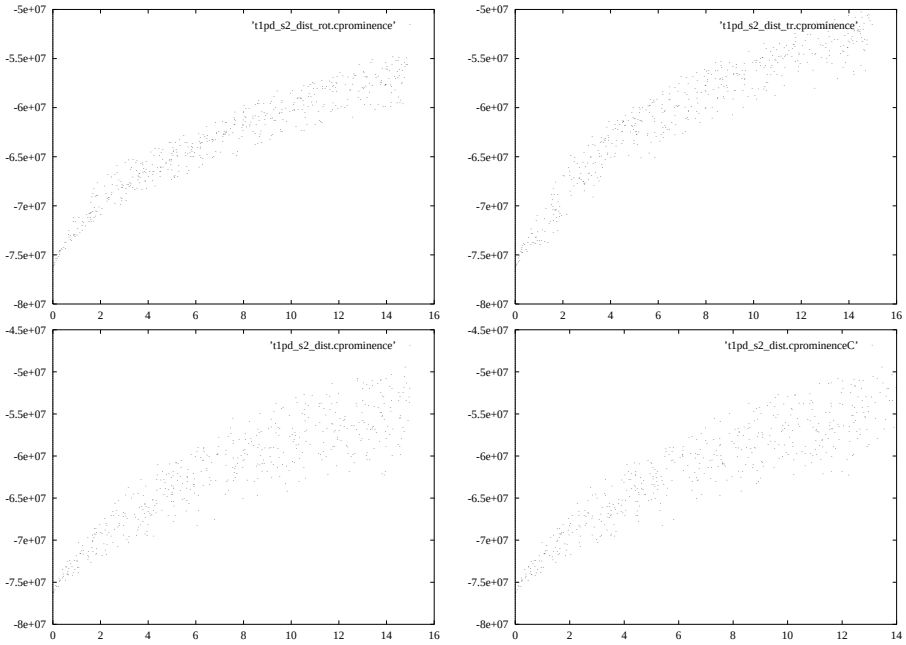

Figure D.10: *Pd/T1: Distance/Cprominence: Rotation only, Translation only, Normal and Corrected.*

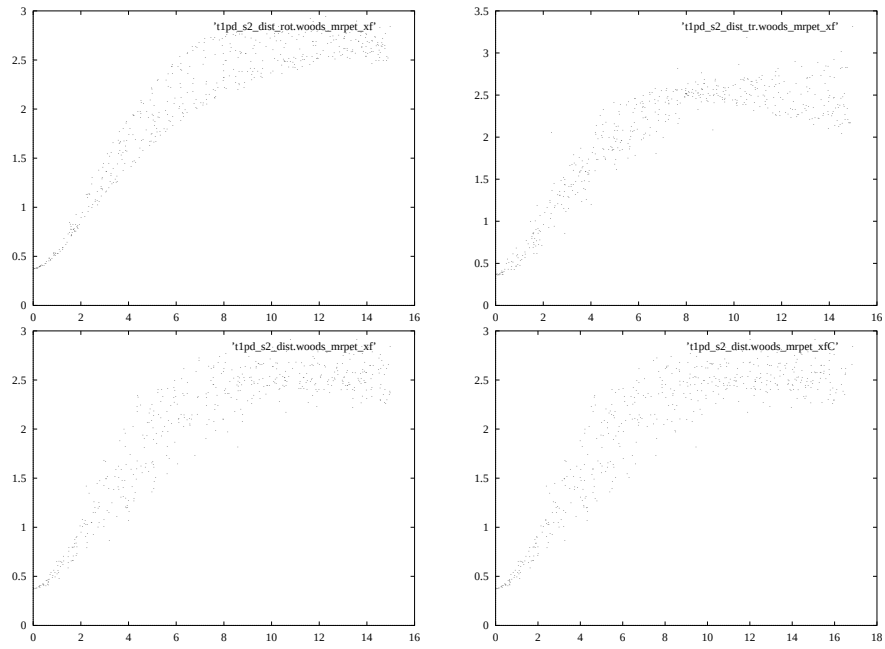

Figure D.11: *Pd/T1: Distance/Woods MR/PET X-fixed: Rotation only, Translation only, Normal and Corrected.*

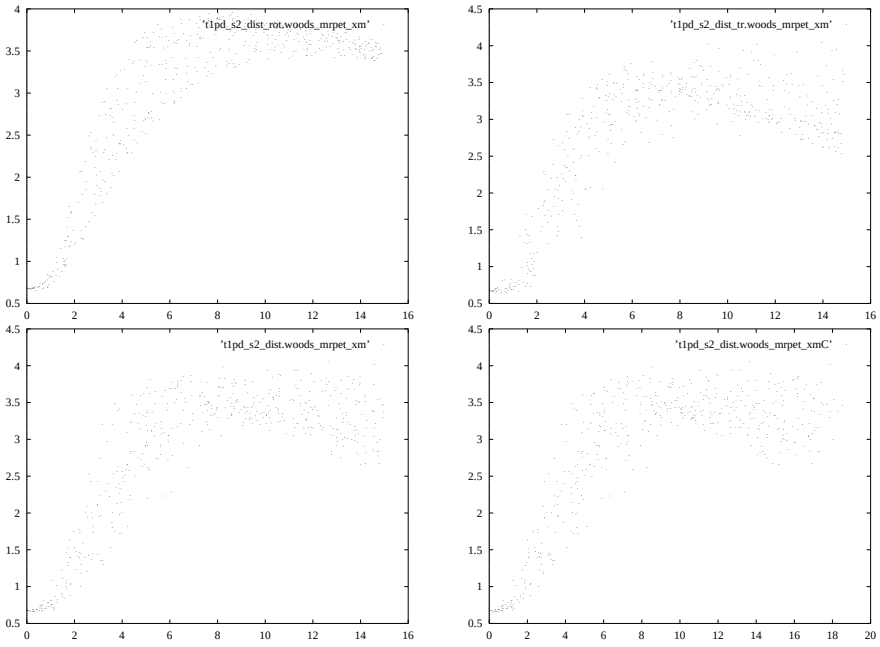

Figure D.12: *Pd/T1: Distance/Woods MR/PET X-moved: Rotation only, Translation only, Normal and Corrected.*

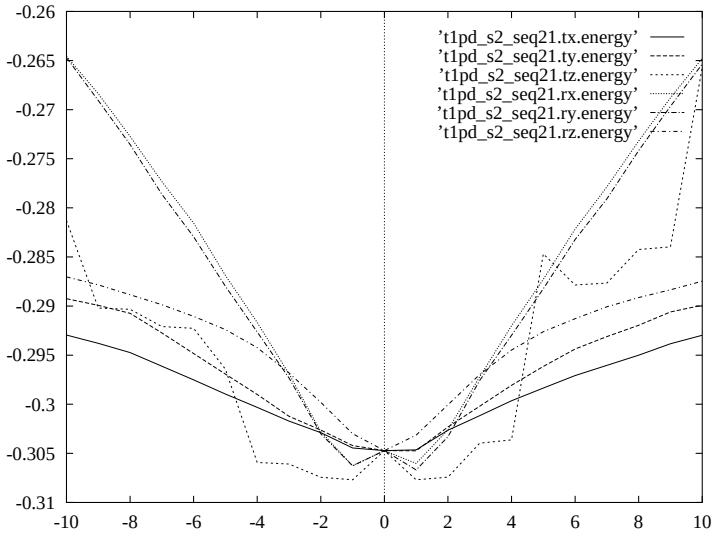

Figure D.13: *Pd/T1: Sequence plots of energy.*

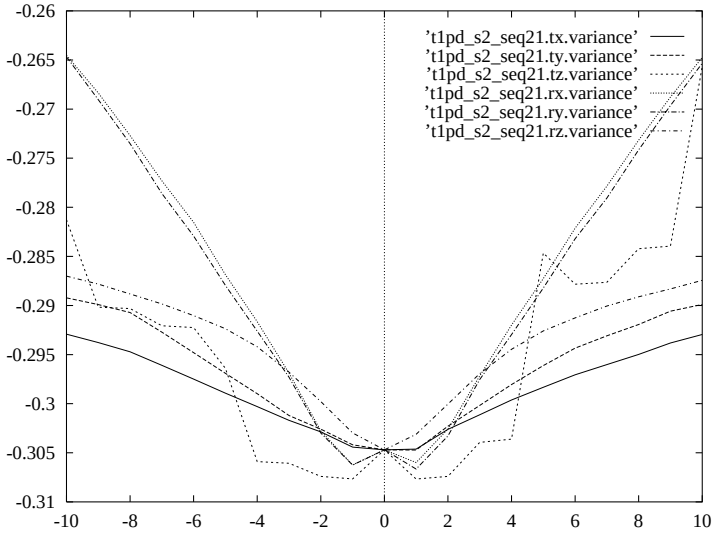

Figure D.14: *Pd/T1: Sequence plots of variance.*

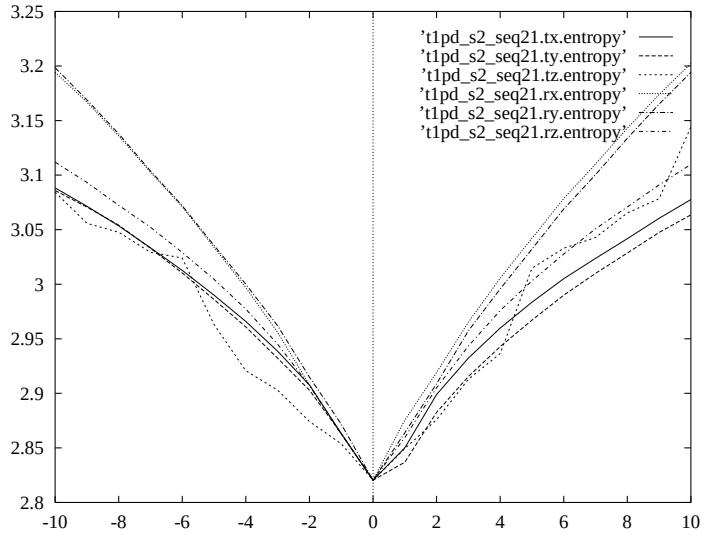

Figure D.15: *Pd/T1: Sequence plots of entropy.*

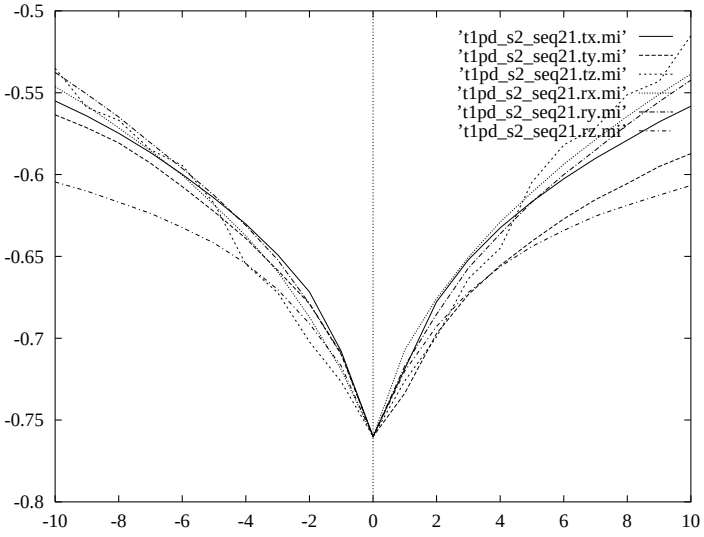

Figure D.16: *Pd/T1: Sequence plots of MI.*

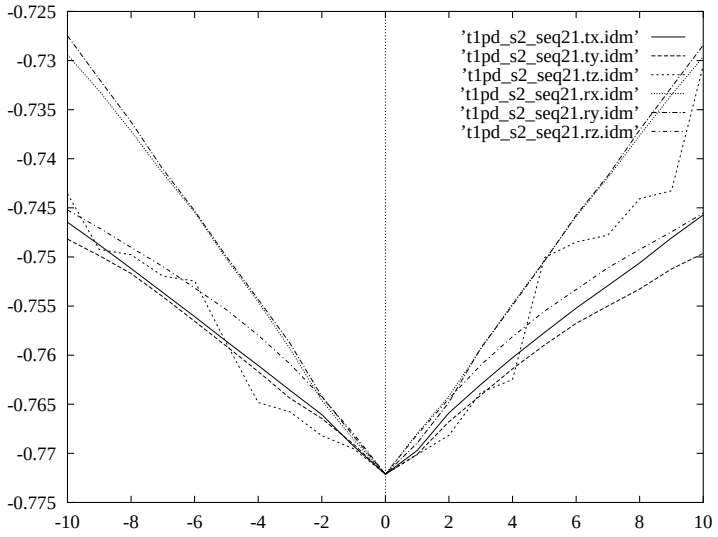

Figure D.17: *Pd/T1: Sequence plots of IDM.*

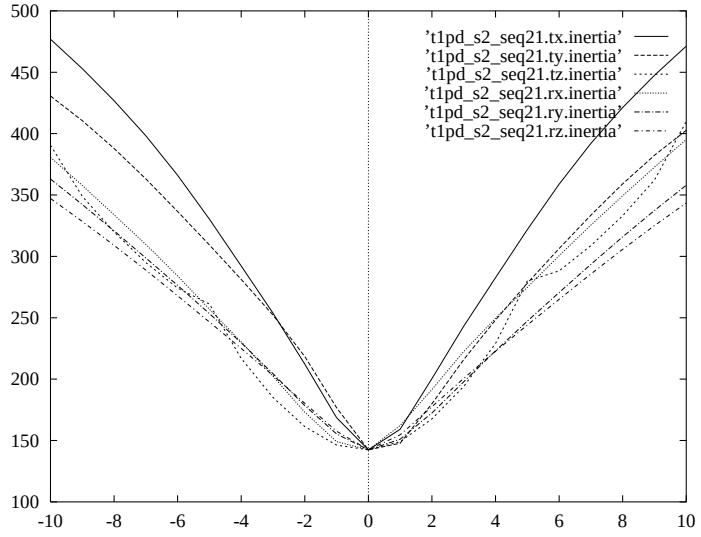

Figure D.18: *Pd/T1: Sequence plots of inertia.*

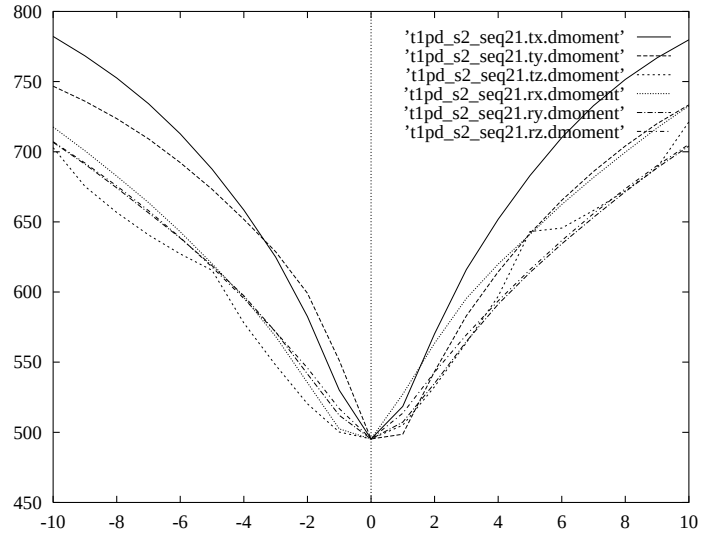

Figure D.19: *Pd/T1: Sequence plots of Dmoment.*

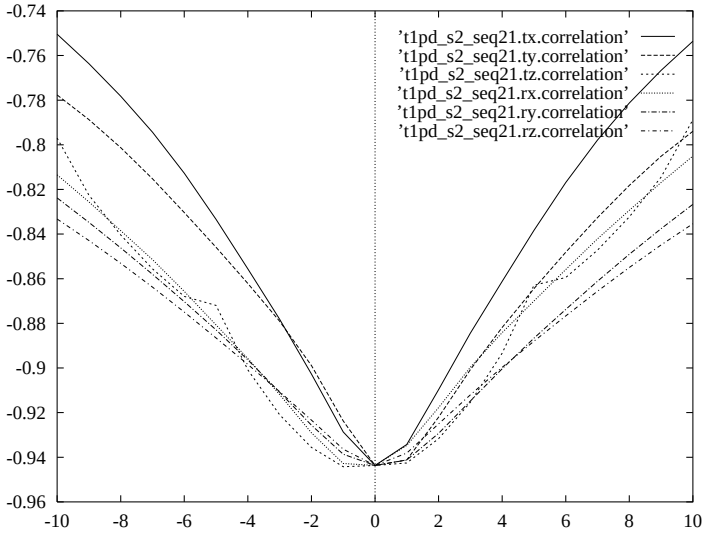

Figure D.20: *Pd/T1: Sequence plots of correlation.*

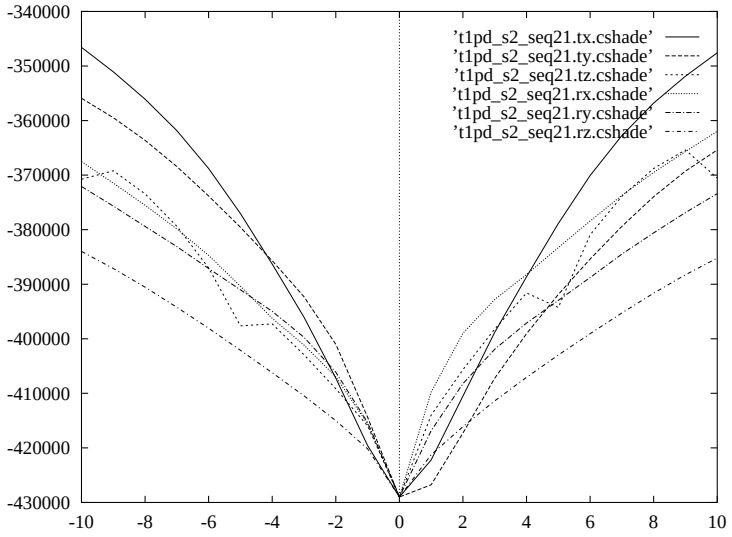

Figure D.21: *Pd/T1: Sequence plots of Cshade.*

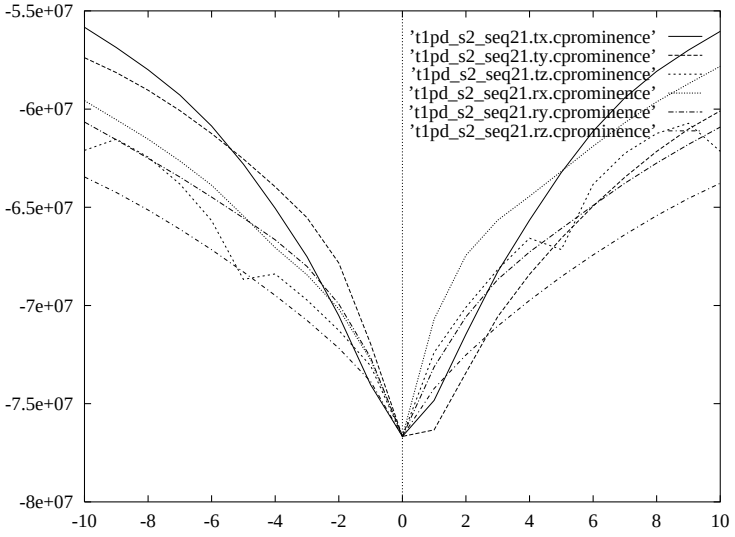

Figure D.22: *Pd/T1: Sequence plots of Cprominence.*

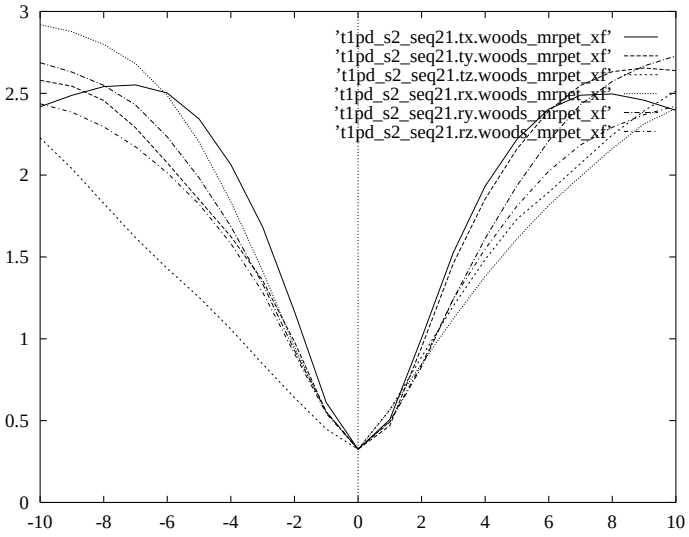

Figure D.23: *Pd/T1: Sequence plots of woods MR/PET X-fixed.*

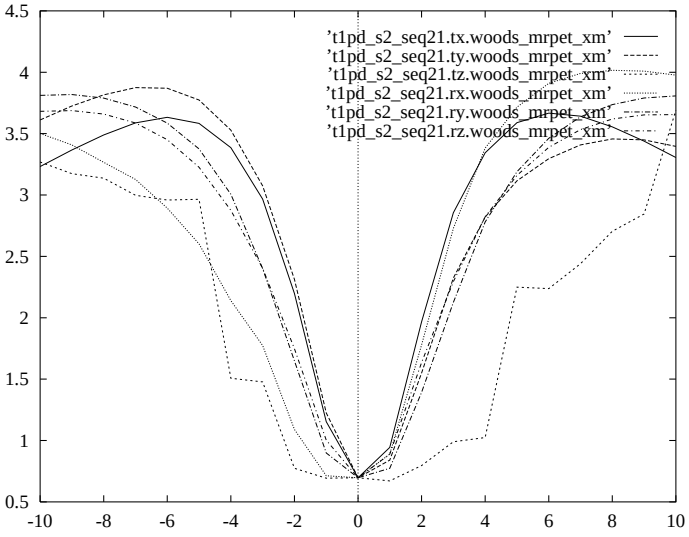

Figure D.24: *Pd/T1: Sequence plots of woods MR/PET X-moved.*

D.2 CT Bone / MR-T1 (moved)

Table D.4: *CT/T1: Regression slope of rotation and translation experiments. Calculated correction factor. 500 samples are used.*

	Rotation	Translation	Correction
Energy	0.002121621	0.002009519	1.0558
Variance	0.002121672	0.002009571	1.0558
Entropy	0.02802454	0.02428523	1.1540
MI	0.00670787	0.0117882	0.5690
IDM	0.00362897	0.003410634	1.0640
Inertia	5.461116	9.80069	0.5572
Dmoment	1.748049	1.489533	1.1736
Correlation	0.001697926	0.005628386	0.3017
Cshade	-690.077	117.8507	NA
Cprominence	-261518.8	-107152.9	2.4406
Woods MR/PET (X:CT)	0.003373523	0.006965458	0.4843
Woods MR/PET (X:T1)	-0.52275	-0.5289265	0.9883

Table D.5: *CT/T1: Corrected linear regression results ($y = ax+b$). 500 samples are used.*

	Slope (a)	Intercept (b)
Energy	0.002057912	-0.266409973
Variance	0.002057961	-0.266383099
Entropy	0.02477265	3.14184302
MI	0.01177012	-0.56200238
IDM	0.003384401	-0.559729930
Inertia	9.502501	417.867211
Dmoment	1.686948	763.647000
Correlation	0.004953982	-0.726158360
Cshade	-280.9123	-179466.4671
Cprominence	-88517.09	-23050940.62
Woods MR/PET (X:CT)	0.006881338	0.511511804
Woods MR/PET (X:T1)	-0.5372226	10.8869138

Table D.6: *CT/T1: Similarity measure plot quality results and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	4	0.9505296	0.9493070
Variance	4	0.9505302	0.9493080
Entropy	5	0.9666757	0.9638251
MI	4	0.8077108	0.7595025
IDM	2	0.9416978	0.9384330
Inertia	1	0.6917992	0.6948765
Dmoment	1	0.2931991	0.2798332
Correlation	1	0.5524907	0.5277128
Cshade	1	0.0773453	0.0773202
Cprominence	1	0.3667449	0.3952463
Woods MR/PET (X:CT)	4	0.6376055	0.6018653
Woods MR/PET (X:T1)	4	0.8147630	0.8139735

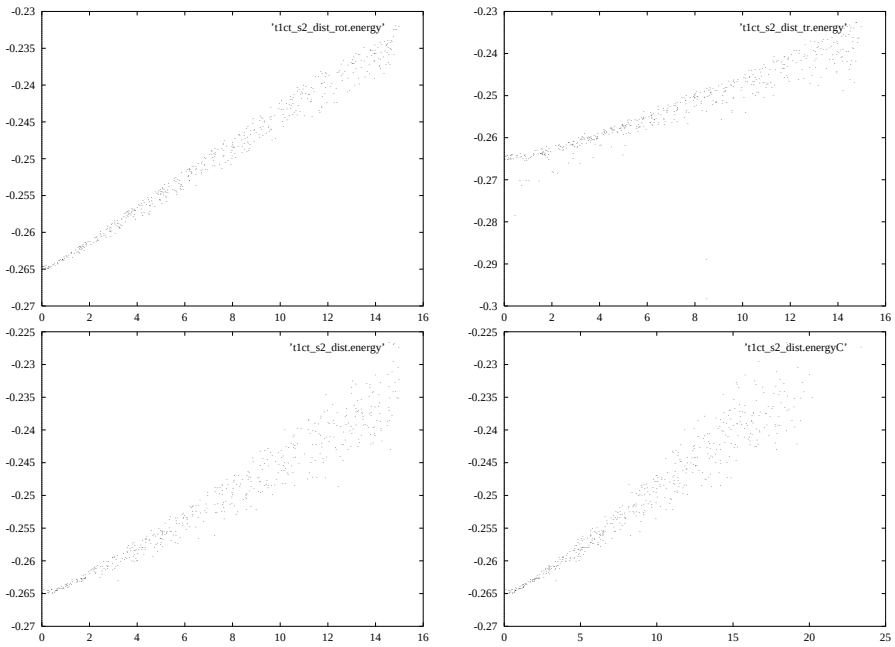

Figure D.25: *CT/T1: Distance/energy: Rotation only, Translation only, Normal and Corrected.*

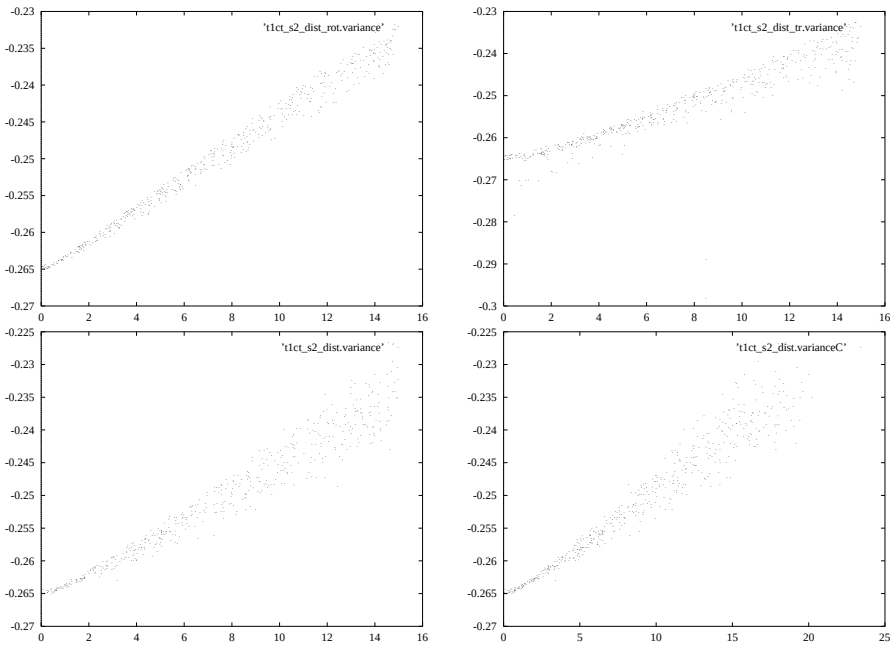

Figure D.26: *CT/T1: Distance/variance: Rotation only, Translation only, Normal and Corrected.*

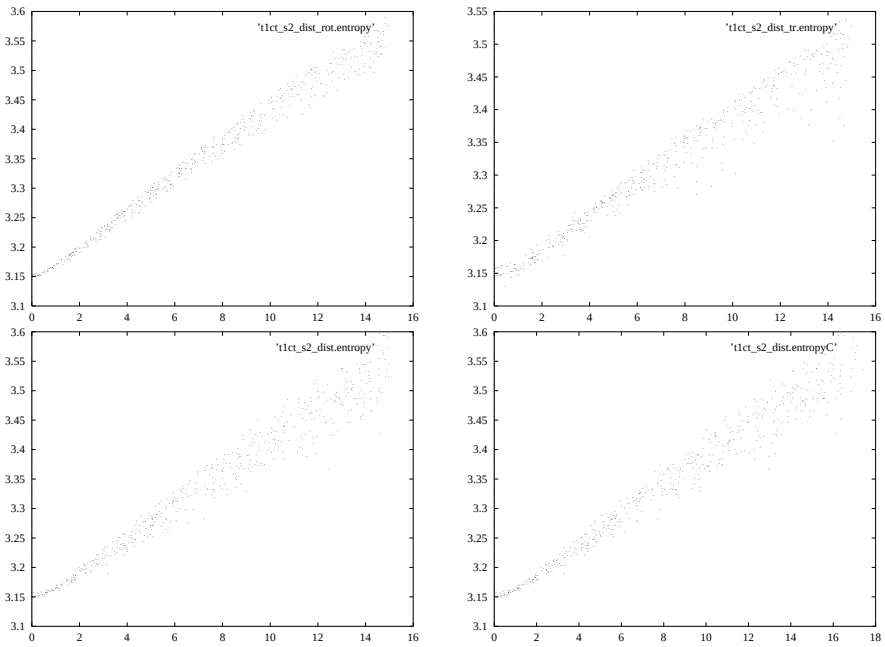

Figure D.27: *CT/T1: Distance/entropy: Rotation only, Translation only, Normal and Corrected.*

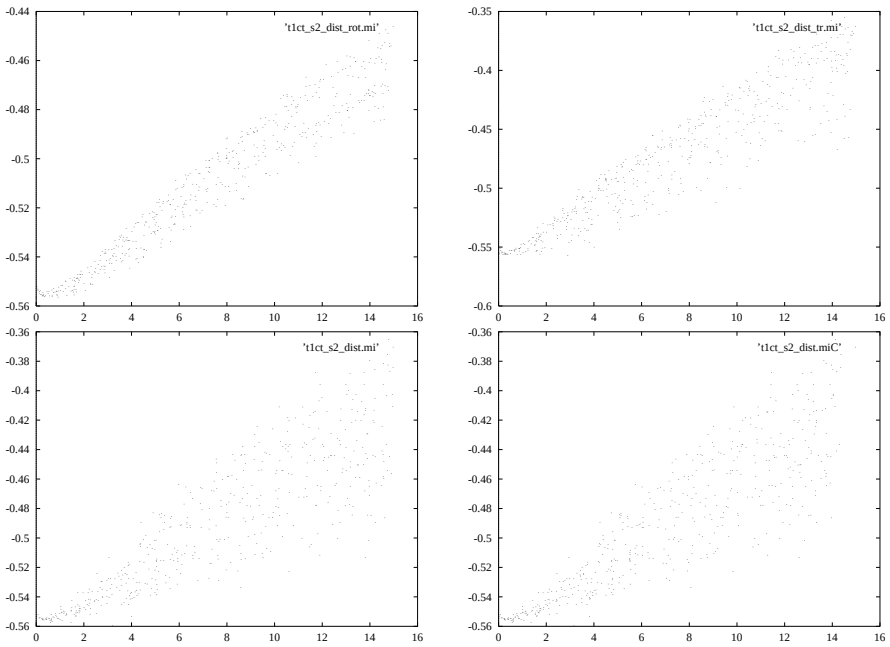

Figure D.28: *CT/T1: Distance/MI: Rotation only, Translation only, Normal and Corrected.*

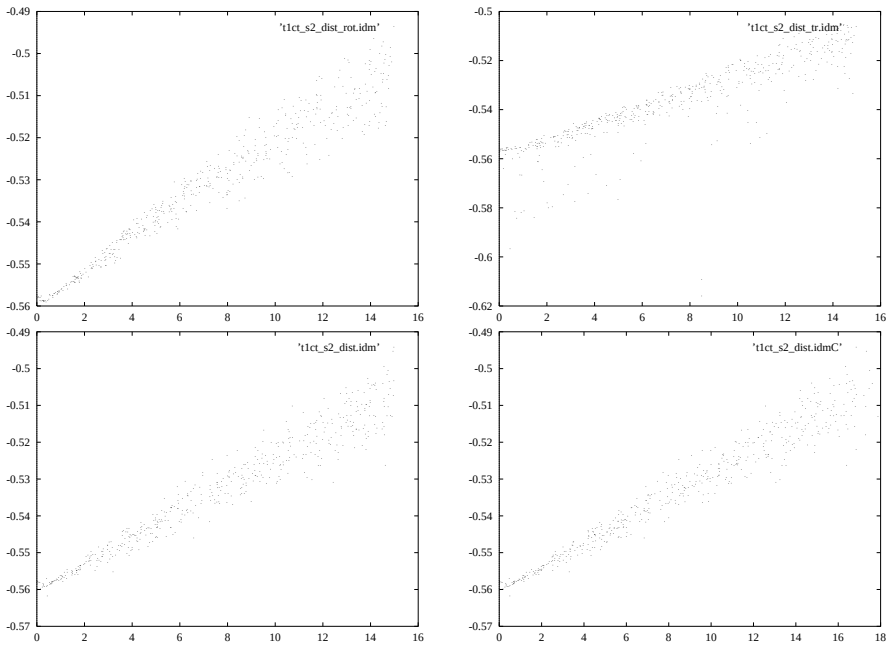

Figure D.29: *CT/T1: Distance/IDM: Rotation only, Translation only, Normal and Corrected.*

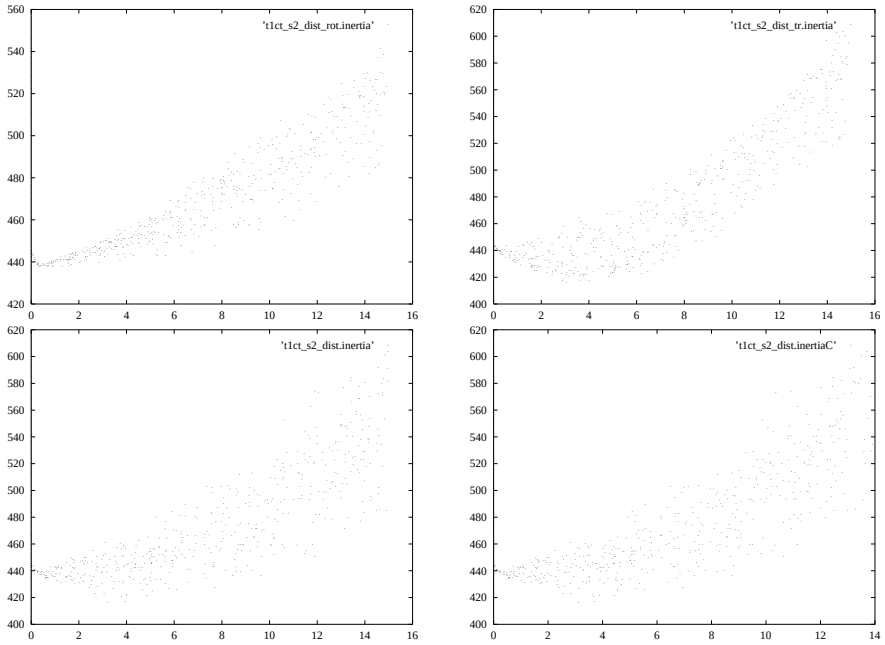

Figure D.30: *CT/T1: Distance/inertia: Rotation only, Translation only, Normal and Corrected.*

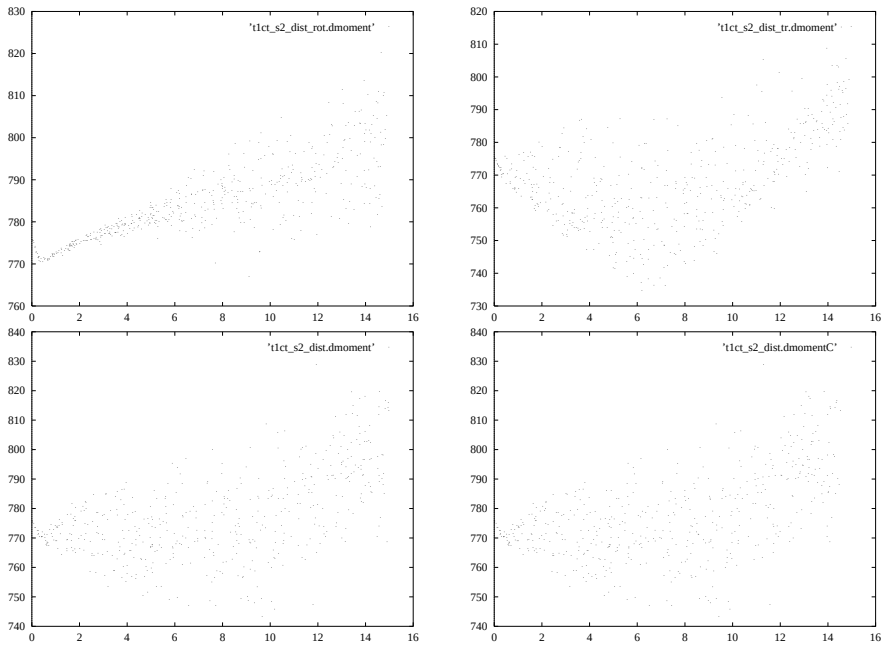

Figure D.31: *CT/T1: Distance/Dmoment: Rotation only, Translation only, Normal and Corrected.*

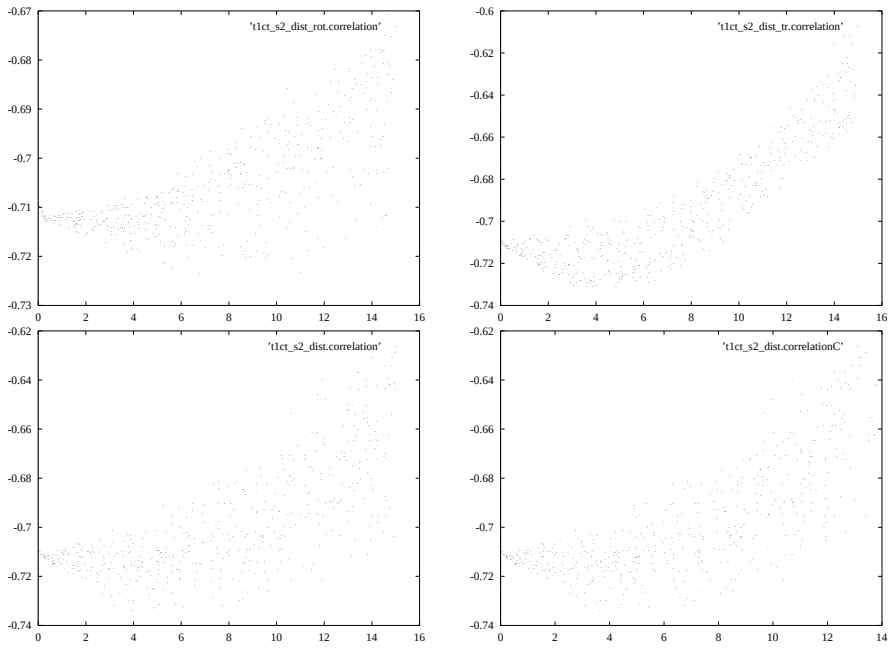

Figure D.32: *CT/T1: Distance/correlation: Rotation only, Translation only, Normal and Corrected. 500 samples are used.*

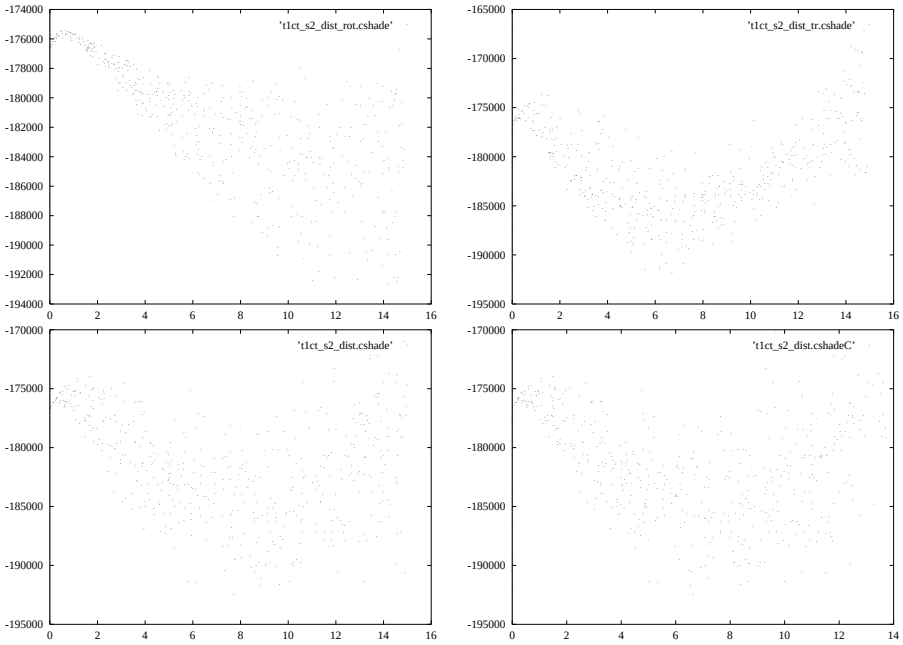

Figure D.33: *CT/T1: Distance/Cshade: Rotation only, Translation only, Normal and Corrected.*

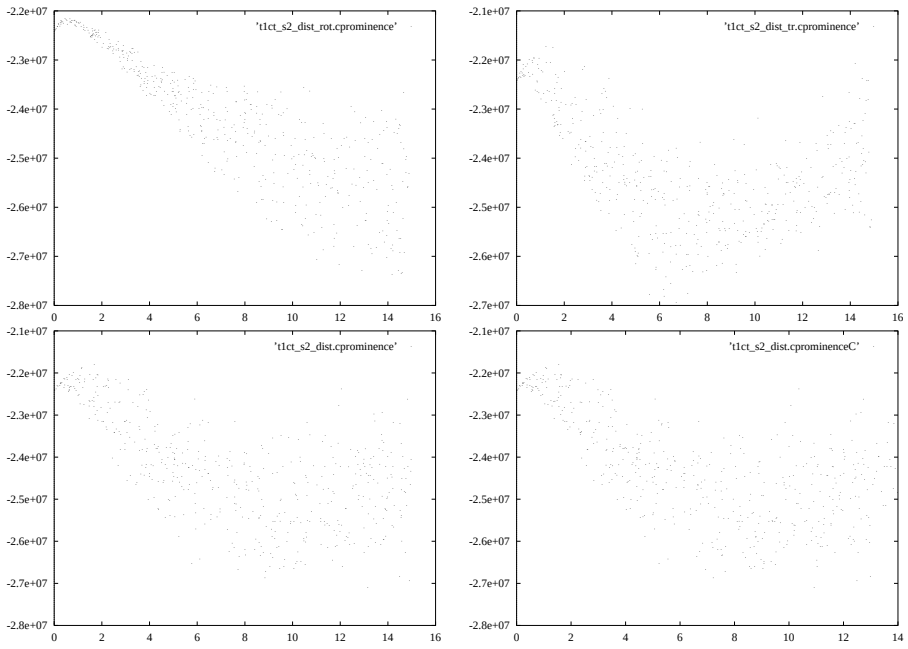

Figure D.34: *CT/T1: Distance/Cprominence: Rotation only, Translation only, Normal and Corrected. 500 samples are used.*

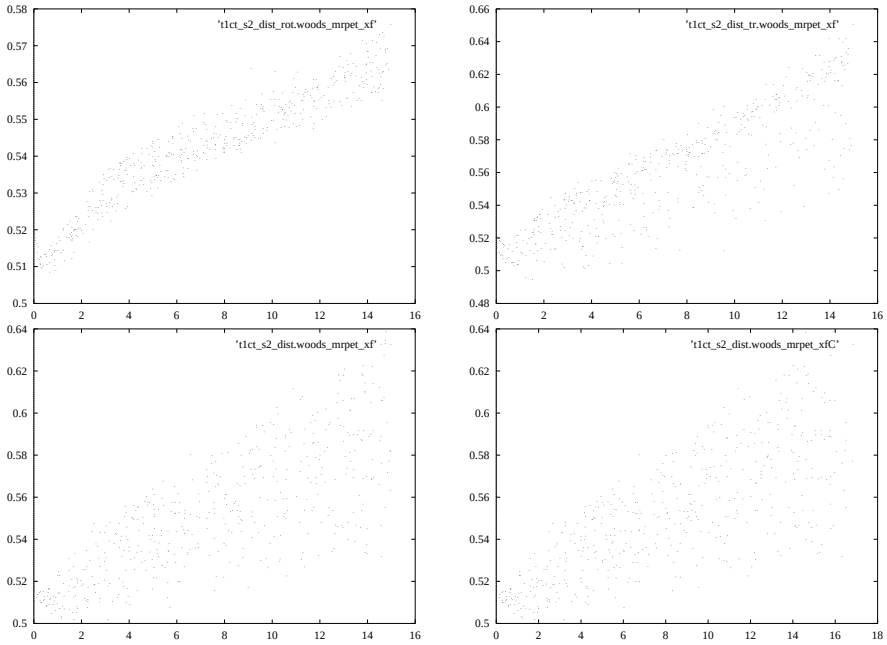

Figure D.35: *CT/T1: Distance/Woods MR/PET X-fixed: Rotation only, Translation only, Normal and Corrected.*

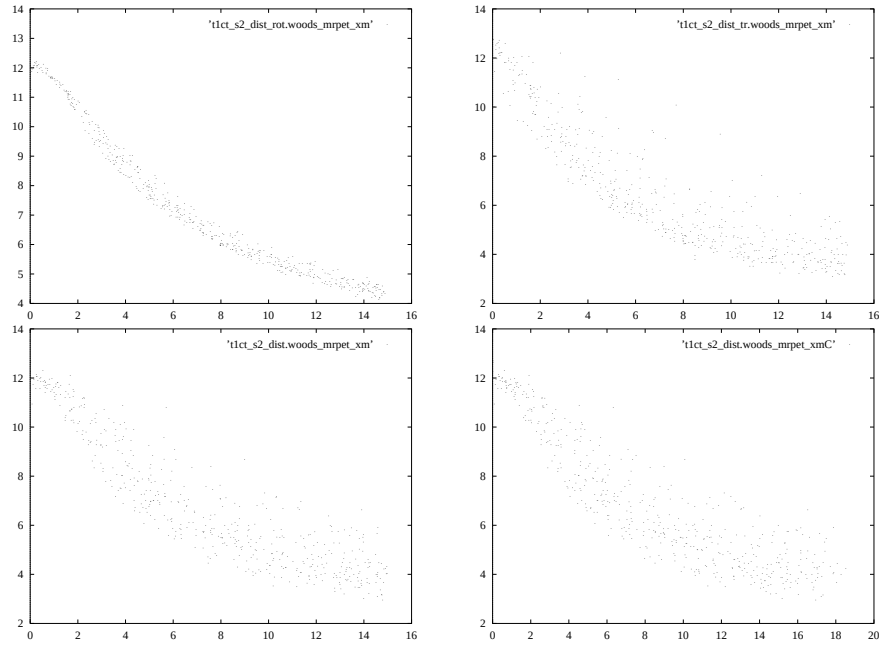

Figure D.36: *CT/T1: Distance/Woods MR/PET X-moved: Rotation only, Translation only, Normal and Corrected.*

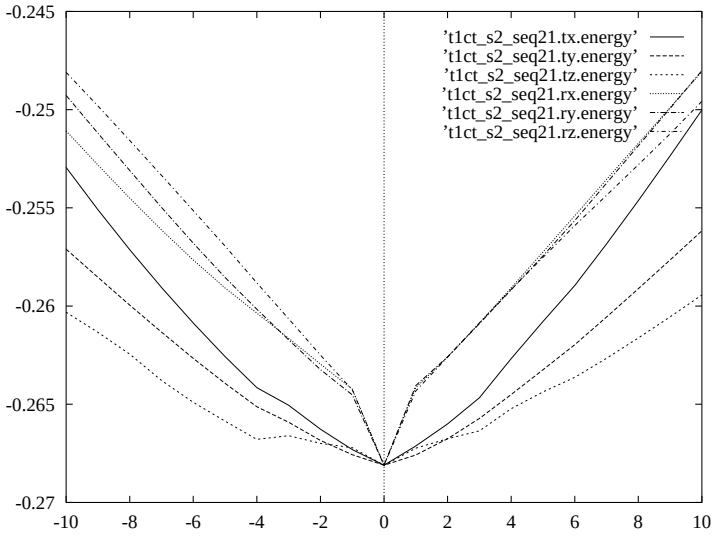

Figure D.37: *CT/T1: Sequence plots of energy.*

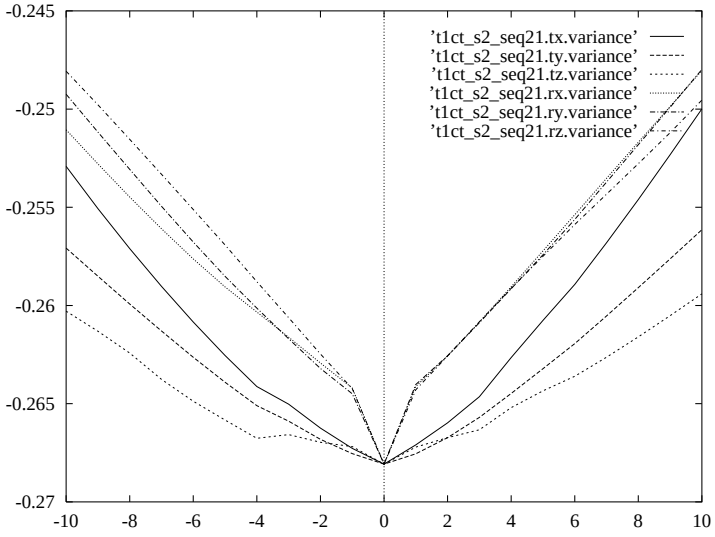

Figure D.38: *CT/T1: Sequence plots of variance.*

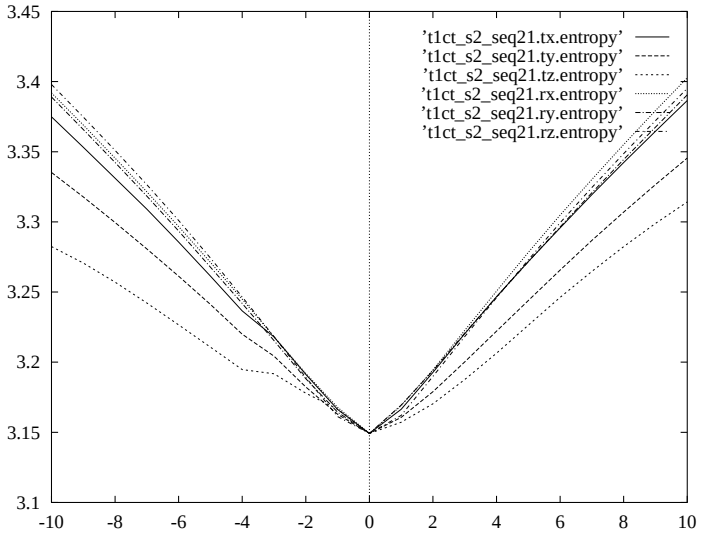

Figure D.39: *CT/T1: Sequence plots of entropy.*

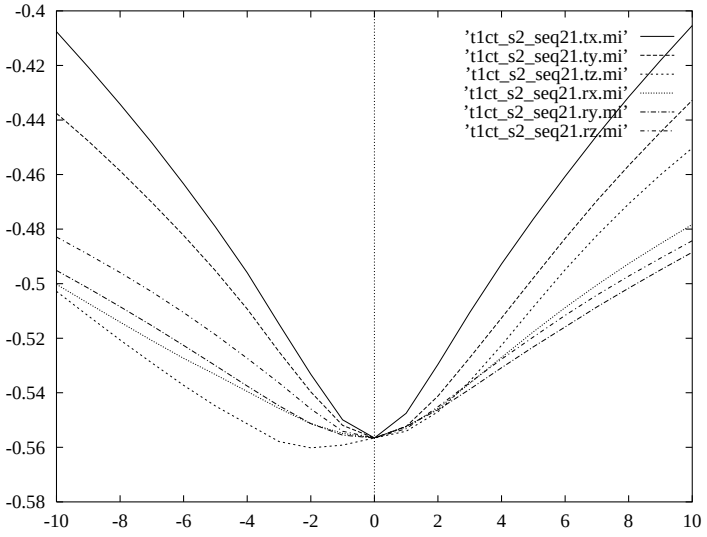

Figure D.40: *CT/T1: Sequence plots of MI.*

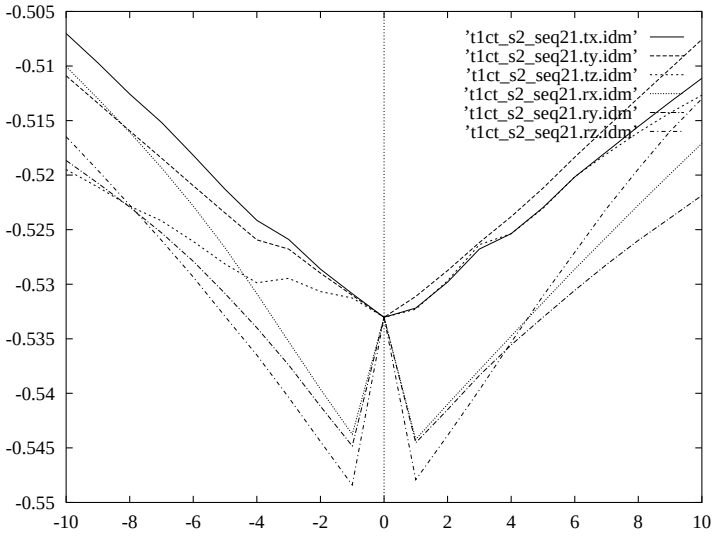

Figure D.41: *CT/T1: Sequence plots of IDM.*

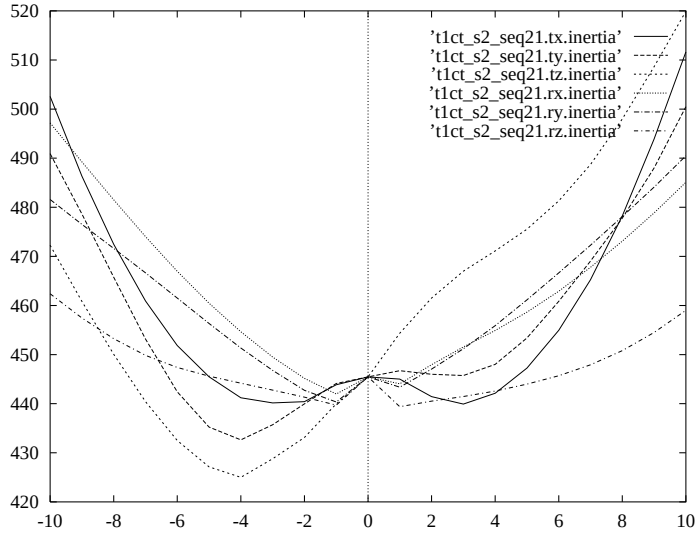

Figure D.42: *CT/T1: Sequence plots of inertia.*

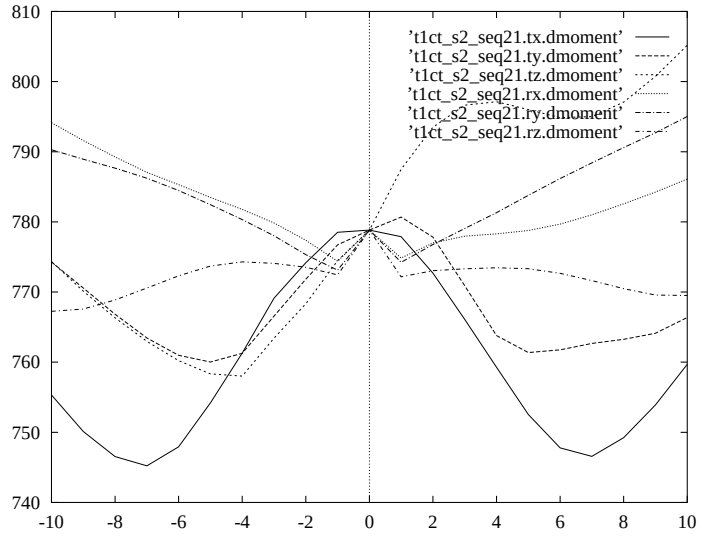

Figure D.43: *CT/T1: Sequence plots of Dmoment.*

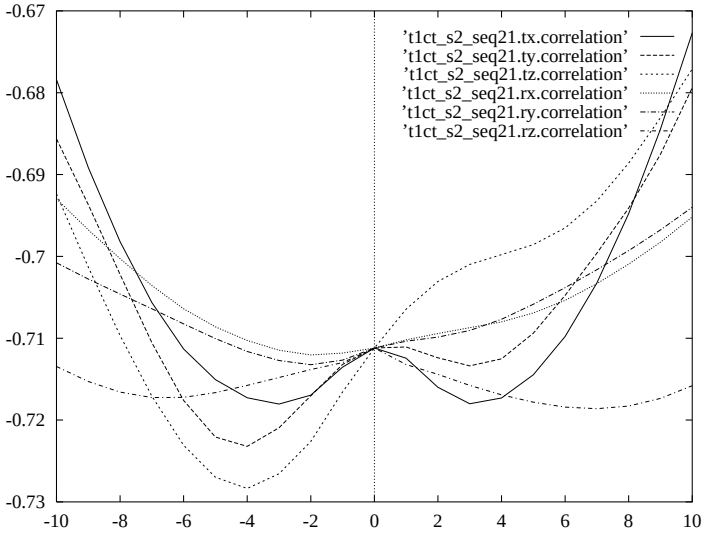

Figure D.44: *CT/T1: Sequence plots of correlation.*

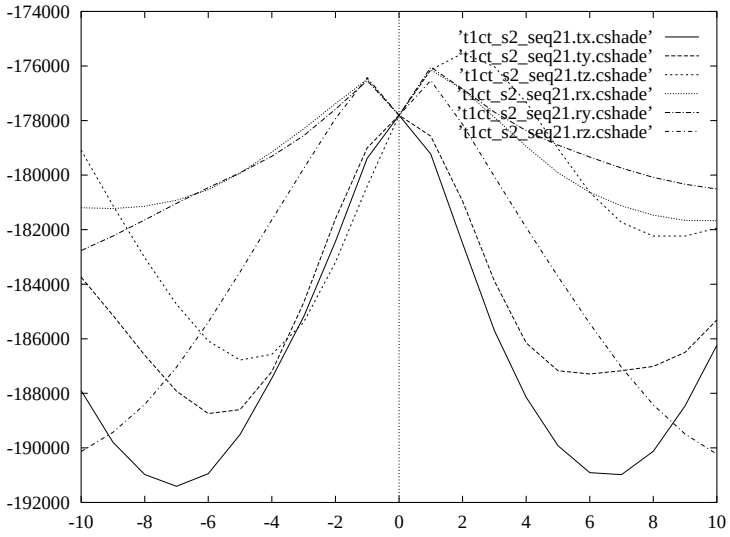

Figure D.45: *CT/T1: Sequence plots of Cshade.*

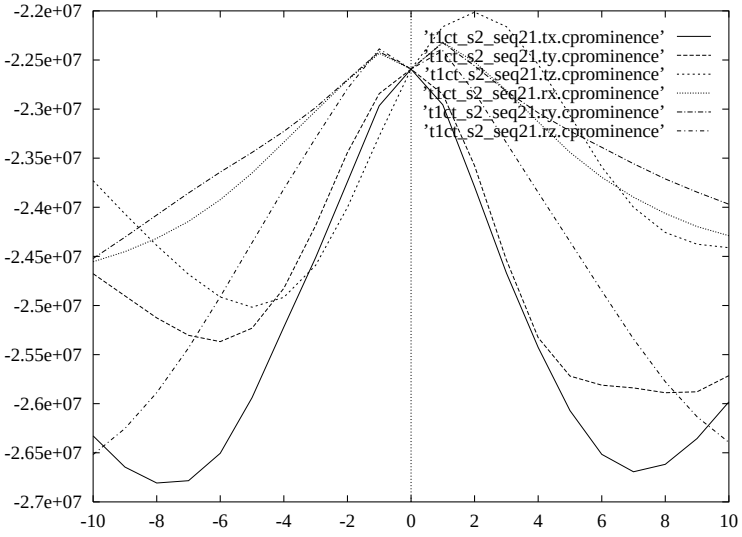

Figure D.46: *CT/T1: Sequence plots of Cprominence.*

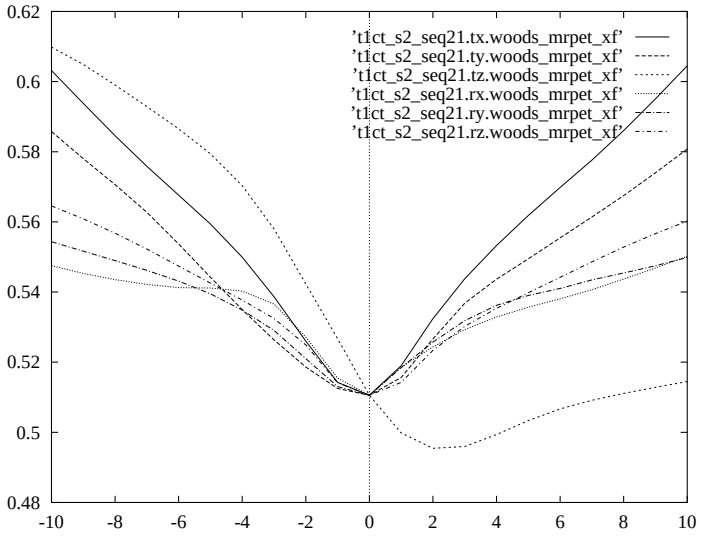

Figure D.47: *CT/T1: Sequence plots of woods MR/PET X-fixed.*

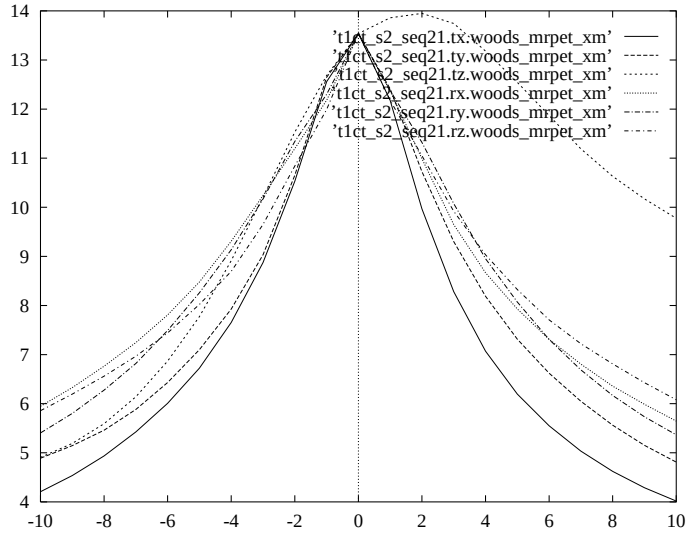

Figure D.48: *CT/T1: Sequence plots of woods MR/PET X-moved.*

D.3 CT Bone (moved) / Red

Table D.7: *CT/R: Regression slope of rotation and translation experiments. Calculated correction factor. 500 samples are used.*

	Rotation	Translation	Correction
Energy	0.0005420492	0.001327608	0.4083
Variance	0.000541966	0.001327518	0.4083
Entropy	0.02588123	0.02403076	1.0770
MI	0.007414116	0.01177257	0.6298
IDM	0.003686641	0.002958455	1.2461
Inertia	41.46096	43.05321	0.9630
Dmoment	25.55221	19.4613	1.3130
Correlation	0.006270843	0.01010701	0.6204
Cshade	2307.93	4829.543	0.4779
Cprominence	654420.1	1230892	0.5317
Woods MR/PET (X:R)	-0.7466082	-0.7829702	0.9536
Woods MR/PET (X:CT)	0.007124646	0.01166874	0.6106

Table D.8: *CT/R: Corrected linear regression results ($y = ax + b$). 500 samples are used.*

	Slope (a)	Intercept (b)
Energy	0.001147005	-0.366643038
Variance	0.001146895	-0.366614460
Entropy	0.02419387	2.81064684
MI	0.01171179	-0.52855416
IDM	0.003042153	-0.497452392
Inertia	43.6012	2478.7321
Dmoment	19.61375	3368.52580
Correlation	0.009952641	-0.849778911
Cshade	4729.952	-915478.910
Cprominence	1233293	-185829490
Woods MR/PET (X:R)	-0.7840236	16.7303990
Woods MR/PET (X:CT)	0.01132891	0.36588970

Table D.9: *CT/R: Similarity measure plot quality results and R^2 compared with uncorrected R^2. 500 samples are used.*

	Quality	R^2	Uncorrected R^2
Energy	1	0.4335973	0.3803990
Variance	1	0.4335405	0.3803409
Entropy	5	0.9801963	0.9790247
MI	4	0.9016652	0.8648146
IDM	3	0.9476440	0.9375267
Inertia	5	0.9721051	0.9725199
Dmoment	5	0.9896430	0.9720567
Correlation	4	0.9176131	0.8917624
Cshade	4	0.8500580	0.7844821
Cprominence	4	0.8491666	0.7947552
Woods MR/PET (X:R)	3	0.7792640	0.7764421
Woods MR/PET (X:CT)	2	0.9013030	0.8753324

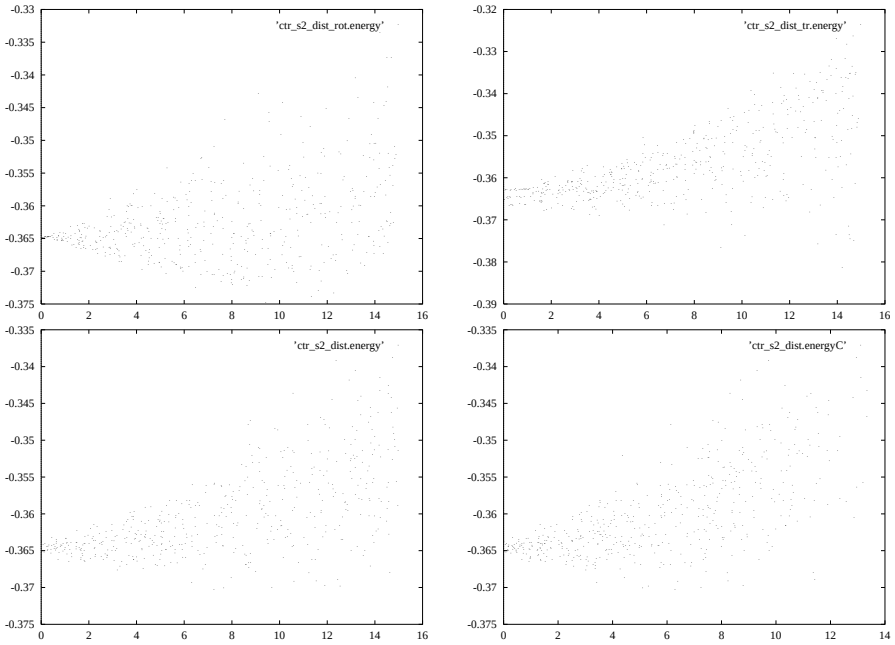

Figure D.49: *CT/R: Distance/energy: Rotation only, Translation only, Normal and Corrected.*

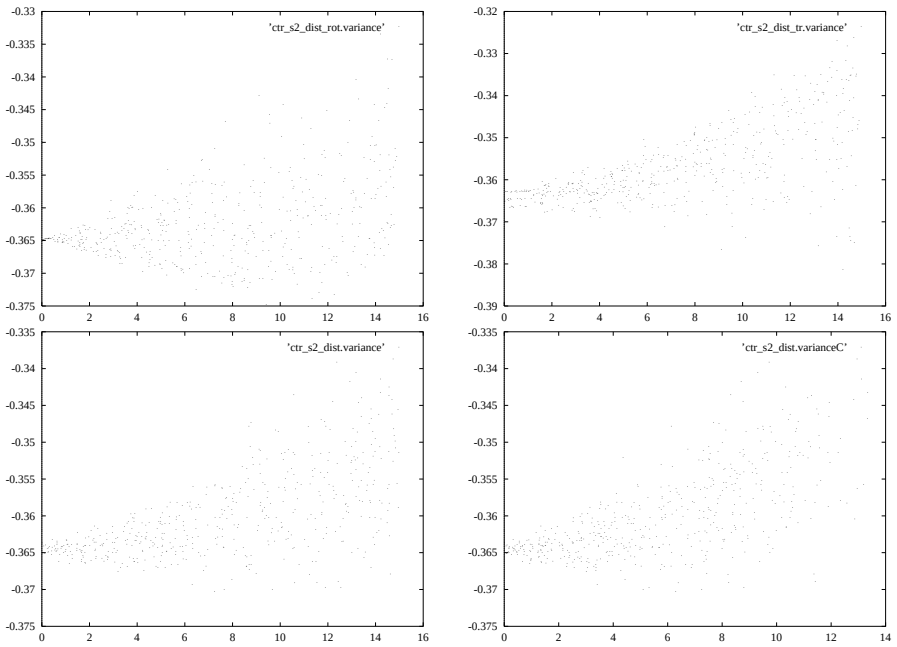

Figure D.50: *CT/R: Distance/variance: Rotation only, Translation only, Normal and Corrected.*

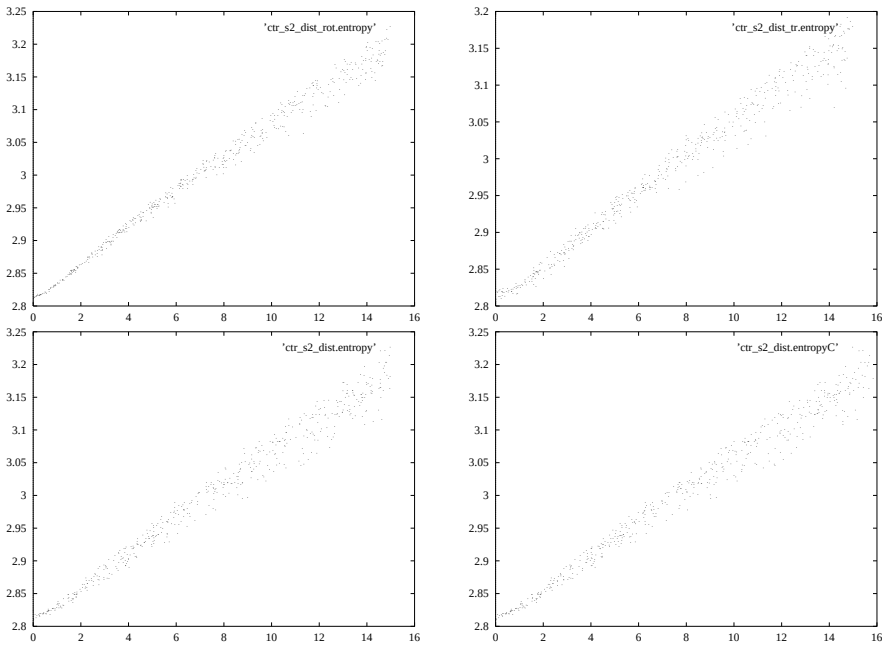

Figure D.51: *CT/R: Distance/entropy: Rotation only, Translation only, Normal and Corrected.*

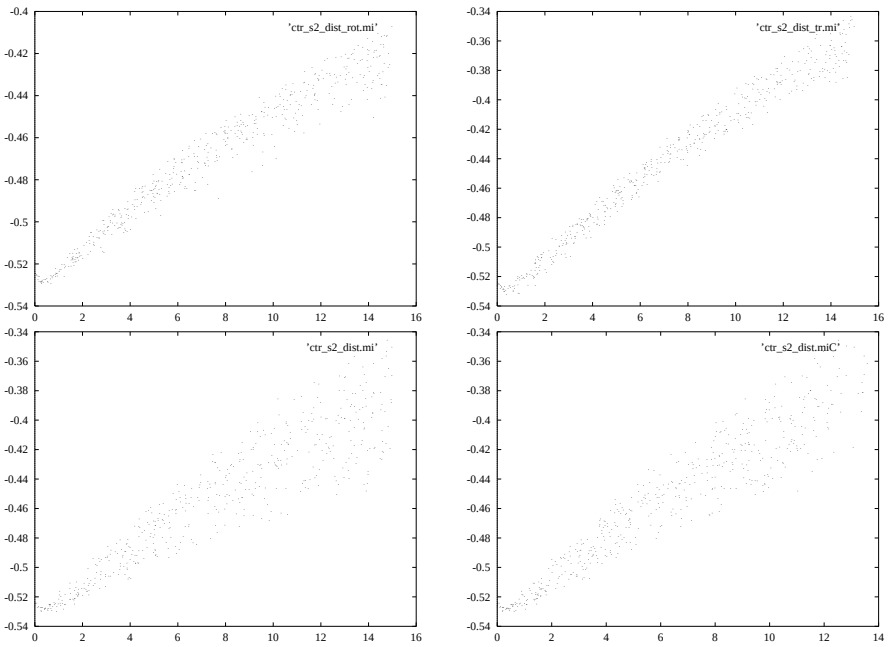

Figure D.52: *CT/R: Distance/MI: Rotation only, Translation only, Normal and Corrected.*

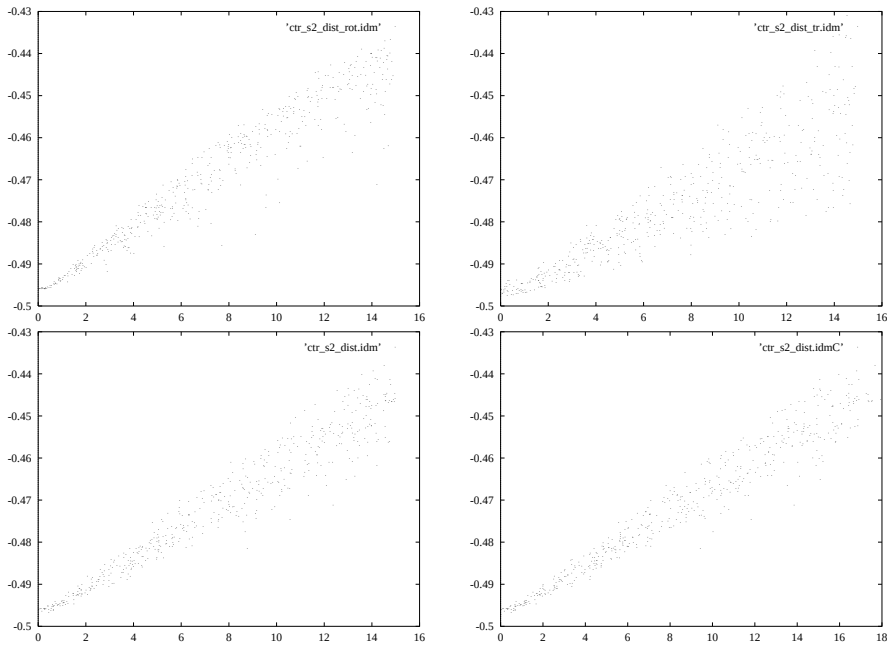

Figure D.53: *CT/R: Distance/IDM: Rotation only, Translation only, Normal and Corrected.*

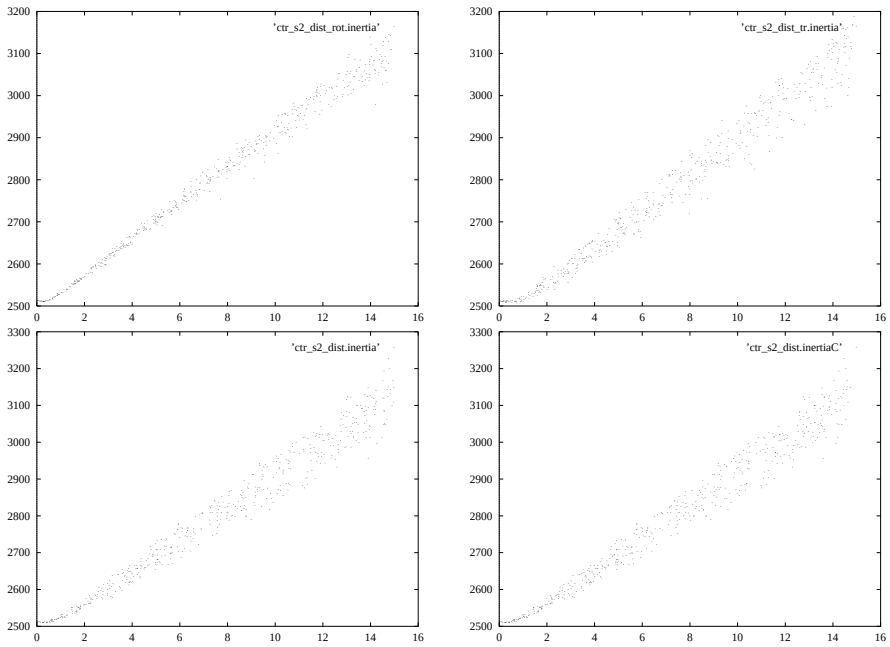

Figure D.54: *CT/R: Distance/inertia: Rotation only, Translation only, Normal and Corrected.*

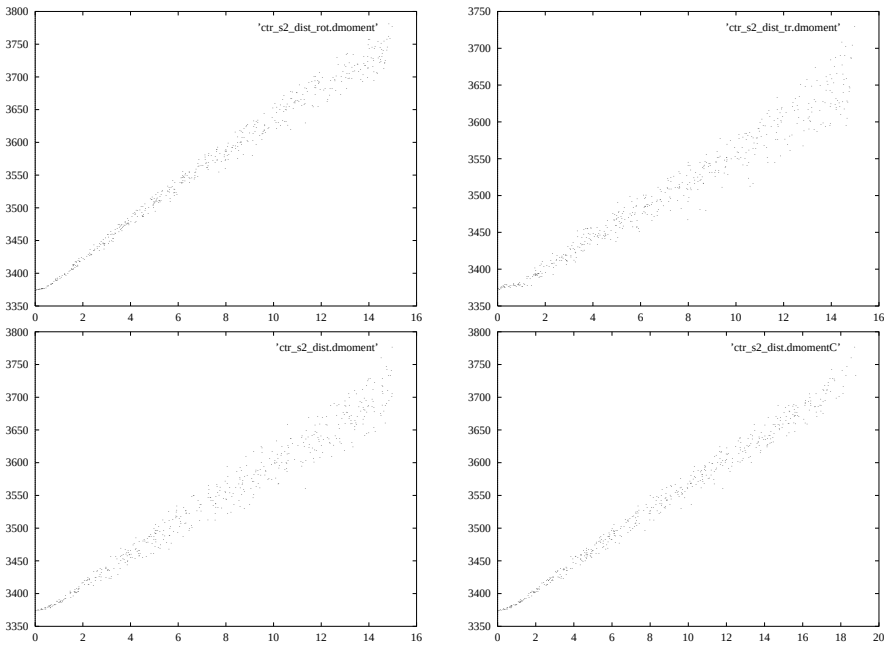

Figure D.55: *CT/R: Distance/Dmoment: Rotation only, Translation only, Normal and Corrected.*

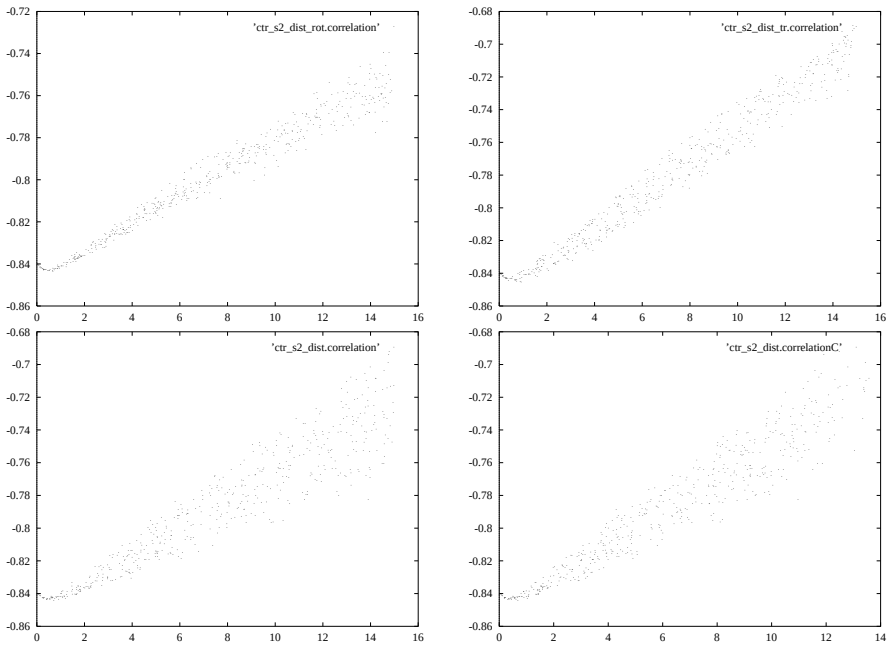

Figure D.56: *CT/R: Distance/correlation: Rotation only, Translation only, Normal and Corrected.*

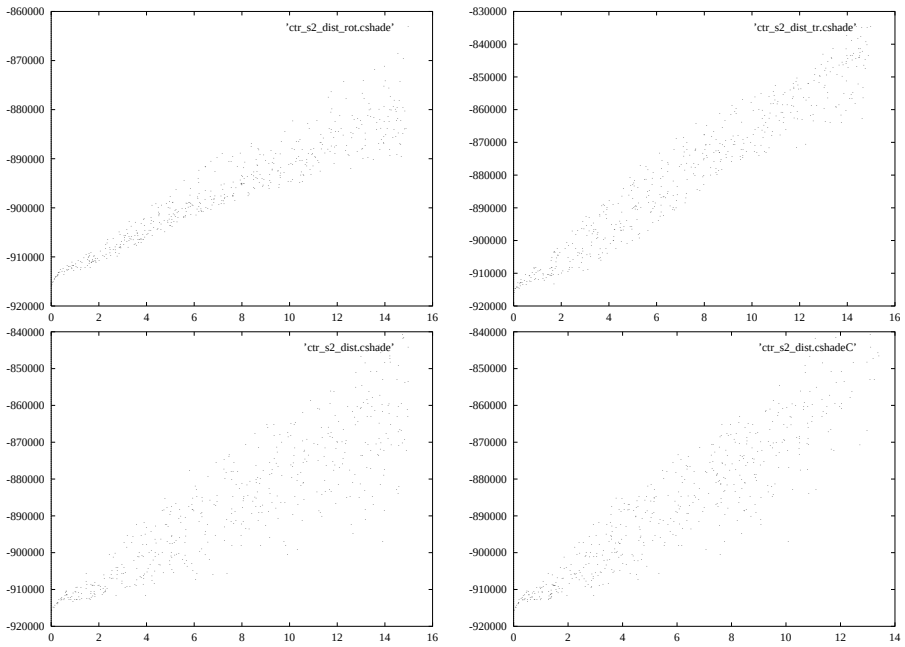

Figure D.57: *CT/R: Distance/Cshade: Rotation only, Translation only, Normal and Corrected.*

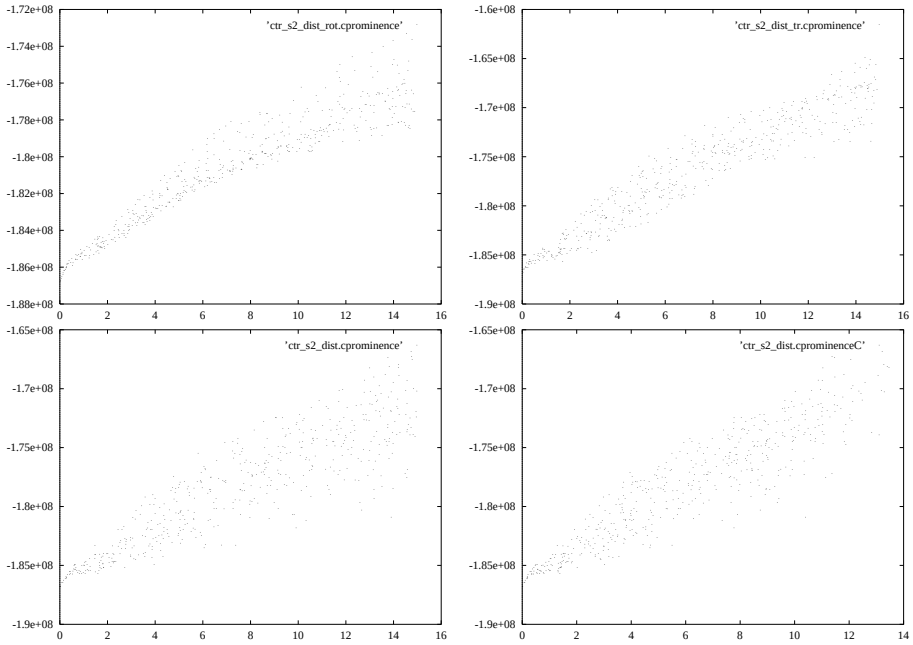

Figure D.58: *CT/R: Distance/Cprominence: Rotation only, Translation only, Normal and Corrected.*

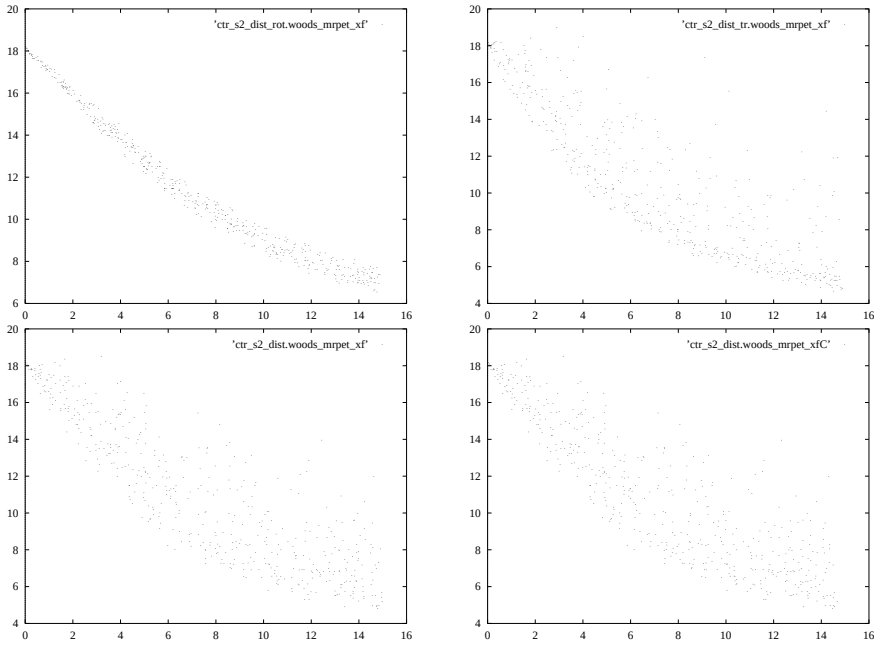

Figure D.59: *CT/R: Distance/Woods MR/PET X-fixed: Rotation only, Translation only, Normal and Corrected.*

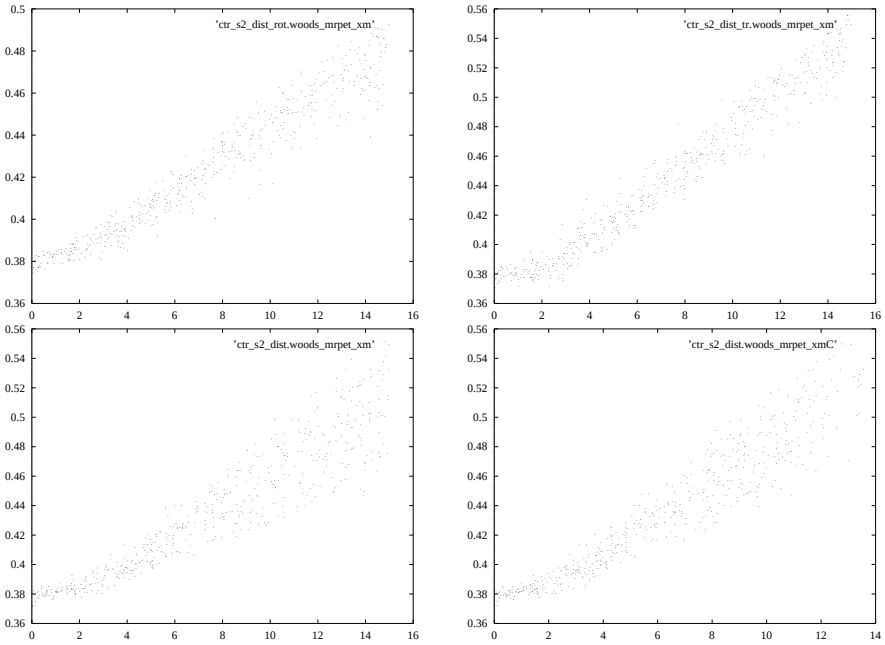

Figure D.60: *CT/R: Distance/Woods MR/PET X-moved: Rotation only, Translation only, Normal and Corrected.*

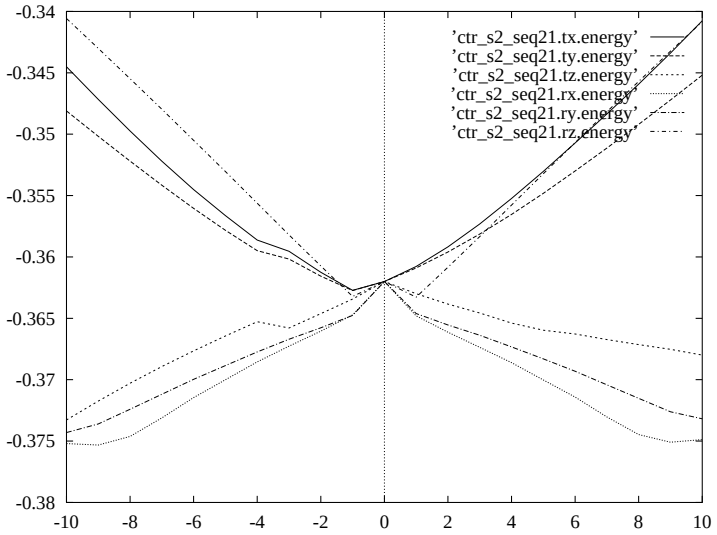

Figure D.61: *CT/R: Sequence plots of energy.*

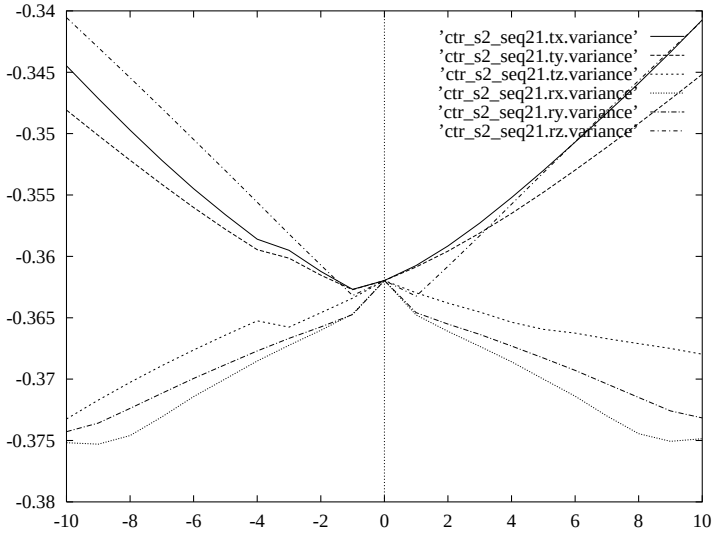

Figure D.62: *CT/R: Sequence plots of variance.*

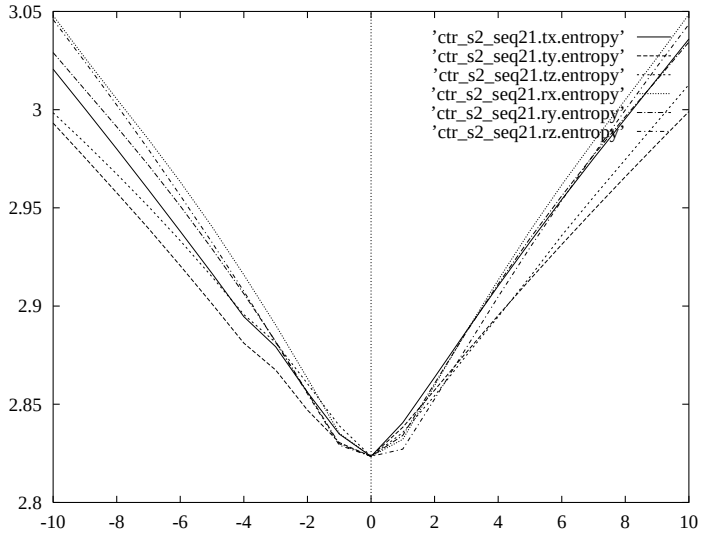

Figure D.63: *CT/R: Sequence plots of entropy.*

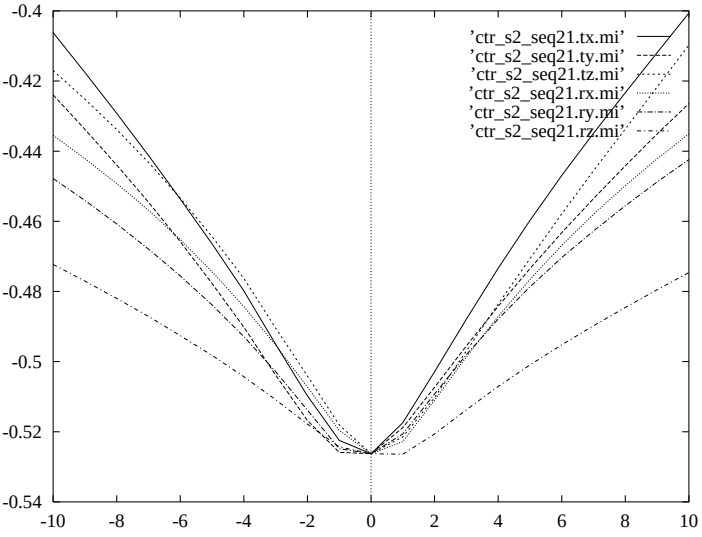

Figure D.64: *CT/R: Sequence plots of MI.*

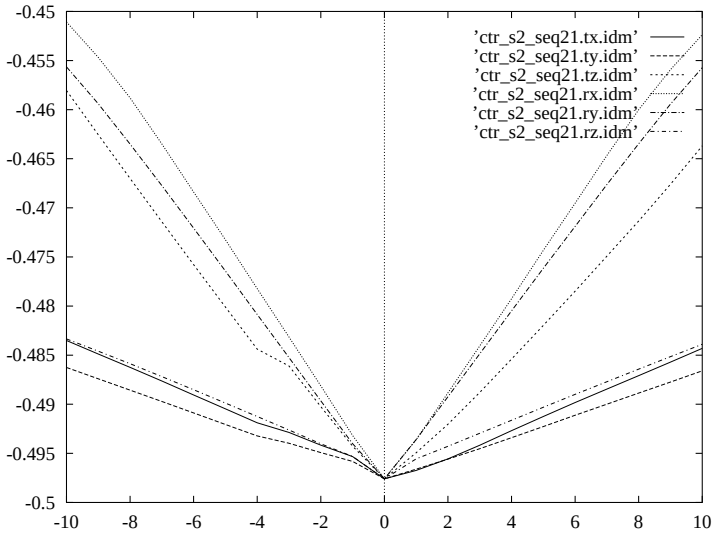

Figure D.65: *CT/R: Sequence plots of IDM.*

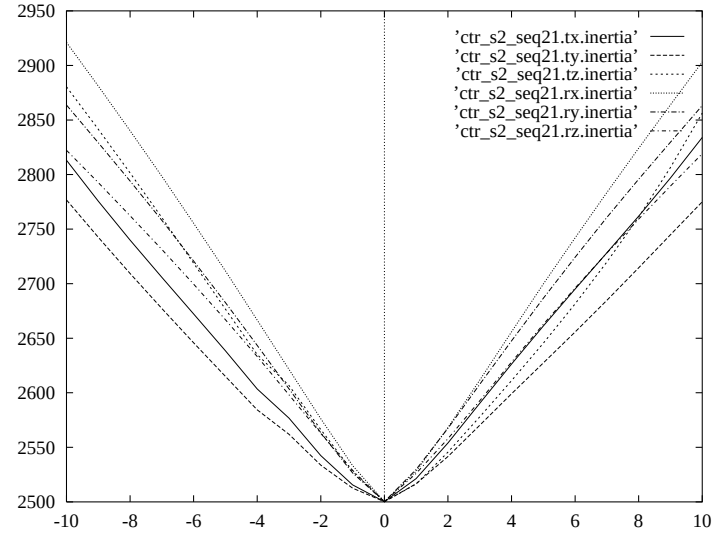

Figure D.66: *CT/R: Sequence plots of inertia.*

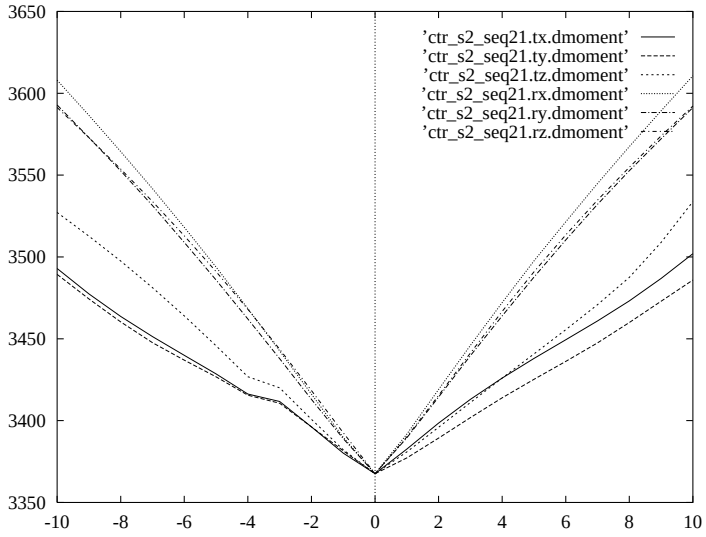

Figure D.67: *CT/R: Sequence plots of Dmoment.*

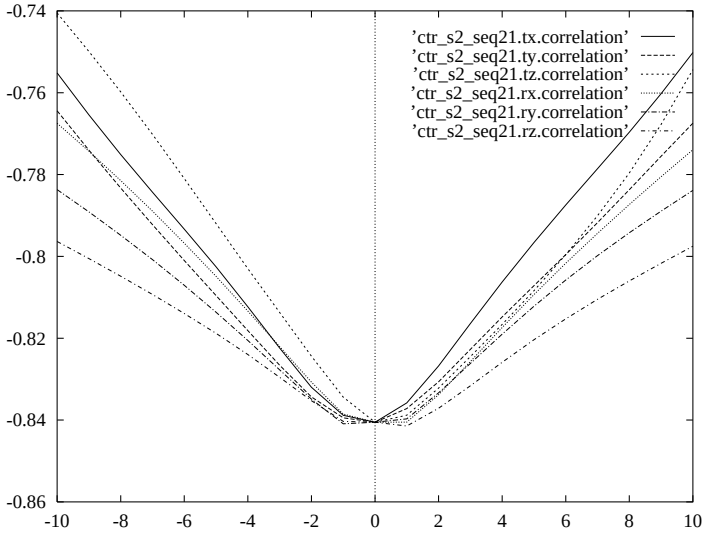

Figure D.68: *CT/R: Sequence plots of correlation.*

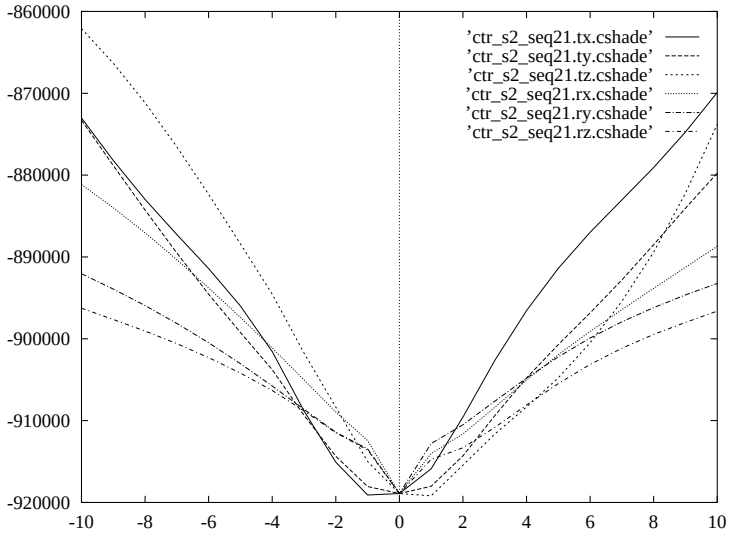

Figure D.69: *CT/R: Sequence plots of Cshade.*

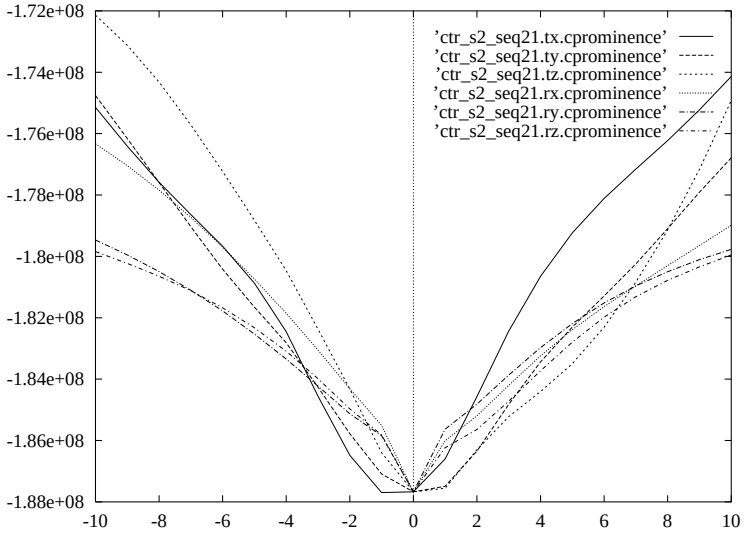

Figure D.70: *CT/R: Sequence plots of Cprominence.*

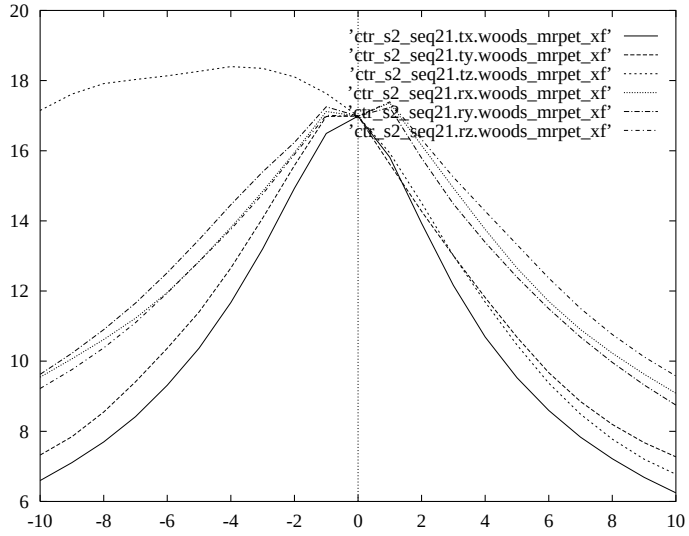

Figure D.71: *CT/R: Sequence plots of woods MR/PET X-fixed.*

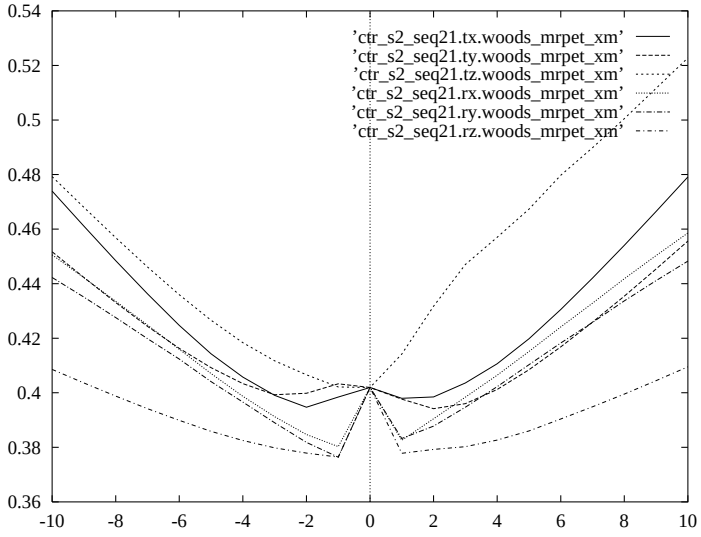

Figure D.72: *CT/R: Sequence plots of woods MR/PET X-moved.*

Bibliography

[Alp90] N.M. Alpert, J.F. Bradshaw, D. Kennedy, and J.A. Correia, *The principal axes transformation - A method for image registration*, J. Nucl. Med., 31:1717-1722, 1990

[Ami91] Y. Amit, U. Grenander, and M. Piccioni, *Structural image restoration through deformable templates*, J. Am. Stat. Ass., 86(414):376-387, 1991

[Api87] A. Apicella, J.S. Kippenham, and J.H. Nagel, *Fast multi-modality image matching*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 9:698-700, 1987

[Ara92] L.K. Arata and A.P. Dhawan, *Iterative principal axes registration: A new algorithm for retrospective correlation of MR-PET brain images*, Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 14:2776-2778, 1992

[Aru87] K.S. Arun, T.S. Huang, and S.D. Blostein, *Least-squares fitting of two 3-D point sets*, IEEE Trans. Pattern Analysis and Machine Intelligence, 9:998-700, 1987

[Baj88] R. Bajcsy, M. Reivich, and S. Kovacic, *Evaluation of registration of PET images and CT images*, Tech. Rep. ms-cis-88-89, grasp lab 163, Depart. of Computer and Information Science, School of Engineering and Applied Science, Univ. of Pennsylvania, 1988

[Baj89] R. Bajcsy and S. Kovacic, *Multiresolution elastic matching*, Computer Vision, Graphics and Image Processing, 46:1-21, 1989

[Brt95] R. Barret et al., *Templates for the solution of linear systems: Building blocks for iterative methods*, WWW http://www.netlib.org/templates/templates.ps, 1995

[Brw77] H.G. Barrow, J.M. Tenembaum, R.C. Bolles, and H.C. Wolf, *Parametric correspondence and chamfer matching*, Proc. 5th Int. Joint Conf. on Artificial Intelligence, Cambridge, MA, pp. 659-663, 1977

[Bar94] K.A. Bartels, A.C. Bovik and C.E. Griffin, *Spatio-temporal tracking of material shape change via multi-dimensional splines*, Proc. IEEE Workshop Biomedical Image Analysis, Seattle, USA, pp. 110-116, June 1994

[Beg88] D. Begis, C. Delpuech, P. le Tallec, L. Loth, and M. Vidrascu, *A finite-element model of tracheal collapse*, the American Physiology Society, pp. 1359-1368, 1988

[Bes92] P.J. Besl and N.D. McKay, *A methods for registration of 3-D shapes*, IEEE Trans. Pattern Analysis and Machine Intelligence, 14(2):239256, 1992

[Bj61] A. Björk, *The relationship of the jaws to the cranium*, in A. Lundstrom, *Introduction to orthodontics*, McGraw-Hill, London, 1961

[Bj63] A. Björk, *Variations in the growth pattern of the human mandible: Longitudinal radiographic study by the implant method*, J. Dent. Res., 42:400, 1963

[Bj69] A. Björk, *Prediction of mandibular growth rotation*, Am. J. Ortho., 55:585, 1969

[Bj72] A. Björk and V. Skieller, *Facial development and tooth eruption*, Am. J. Ortho., 62:339, 1972

[Bj77] A. Björk and V. Skieller, *Growth of the maxilla in three dimensions as revealed radiographically by the implant method*, Brit. J. Ortho., 4:53, 1977

[Boe90] R. Boesecke, T. Bruckner, and G. Ende, *Landmark based correlation of medical images*, Phys. Med. Biol., 35(1):121-126, 1990

[Boo89] F.L. Bookstein, *Principal warps: Thin-plate splines and the decomposition of deformations*, IEEE Trans. Pattern Analysis and Machine Intelligence, 11:567-585, 1989

[Boo92] F.L. Bookstein and W.D.K. Green, *Edge information at landmarks in medical images*, Proc. SPIE 1808: Visualization in Biomedical Computing, pp. 242-258, 1992

[Bor88] G. Borgefors, *Hierarchical chamfer matching: A paramteric edge matching algorithm*, IEEE Trans. Pattern Analysis and Machine Intelligence, 10:849-865, 1988

[BN94] M. Bro-Nielsen, *Active nets and cubes*, IMM Tech. Rep., 1994

[BN95] M. Bro-Nielsen, *Modelling elasticity in solids using Active Cubes - Application to simulated operations*, Proc. Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), pp. 535-541, 1995

[BN95b] M. Bro-Nielsen and S. Cotin, *Soft tissue modelling in surgery simulation for prediction of results of craniofacial operations & steps towards virtual reality training systems*, Proc. 3rd Int. Workshop on Rapid Prototyping in Medicine & Computer-Assisted Surgery, 1995

[BN96] M. Bro-Nielsen, *Mvox: Interactive 2-4D medical image and graphics visualization software*, Proc. Computer Assisted Radiology (CAR'96), 1996

[BN96b] M. Bro-Nielsen, *Mvox WWW homepage*, http://www.imm.dtu.dk/ mvox/, 1996

[BN96c] M. Bro-Nielsen and S. Cotin, *Real-time volumetric deformable models for surgery simulation using finite elements and condensation*, accepted for Eurographics'96, 1996

[BN96d] M. Bro-Nielsen, *Surgery simulation using fast finite elements*, accepted for VBC'96, 1996

[BN96e] M. Bro-Nielsen and C. Gramkow, *Fast fluid registration of medical images*, accepted for Visualization in Biomedical Imaging (VBC'96), 1996

[Brt81] C. Broit, *Optimal registration of deformed images*, Doctoral dissertation, University of Pennsylvania, August 1981

[Bro92] L.G. Brown, *A survey of image registration techniques*, ACM Computing Surveys, 24(4):325-376, 1992

[Bur81] D.J. Burr, *A Dynamic model for image registration*, Computer Vision, Graphics, and Image Processing, 15:102-112, 1981

[Cap93] L. Caponetti and A.M. Fanelli, *Computer-aided simulation for bone surgery*, IEEE Computer Graphics & Applications, November, pp. 86-92, 1993

[Car86] G.F. Carey and J.T. Oden, *Finite elements: Fluid mechanics Volume VI*, Prentice-Hall, New Jersey, ISBN 0-13-317132-9 025, 1986

[Che88] C-T. Chen, C.A. Pelizarri, G.T.Y. Chen, M.D. Cooper, and D.N. Levin, *Image analysis of PET data with the aid of CT and MR images*, Proc. Information Processing in Medical Imaging, pp. 601-611, 1988

[Che92] D.T. Chen and D. Zeltzer, *Pump it up: Computer animation of a biomechanically based model of muscle using the finite element method*, Computer Graphics, 26(2):89-98, 1992

[Chr93] G.E. Christensen, R.D. Rabbitt, and M.I. Miller, *A deformable neuroanatomy textbook based on viscous fluid mechanics*, Proc. Ann. Conf. Information Sciences and Systems, pp. 211-216, 1993

[Chr94] G.E. Christensen, M.I. Miller and M. Vannier, *A 3D deformable magnetic resonance textbook based on elasticity*, in AAAI Spring Symposium Series: Applications of Computer Vision in Medical Image Processing, pp. 153-156, Stanford University, March 1994

[Chr94b] G.E. Christensen, R.D. Rabbitt and M.I. Miller, *3D brain mapping using a deformable neuroanatomy*, Physics in Medicine and Biology, 39:609-618, 1994

[Chr94c] G.E. Christensen, *Deformable shape models for anatomy*, Washington University Ph.D. Thesis, August 1994

[Chr96] G.E. Christensen, M.I. Miller, M. Vannier and U. Grenander, *Individualizing neuroanatomical atlases using a massively parallel computer*, IEEE Computer, 29(1):32-38, January 1996

[Chr96b] G.E. Christensen, A.A. Kane, J.L. Marsh, and M. Vannier, *A 3D deformable infant atlas*, Proc. Computer Assisted Radiology (CAR'96), pp. 847-852, 1996

[Cia87] P.G. Ciarlet, *Mathematical elasticity Vol. 1: Three-dimensional elasticity*, North-Holland, Amsterdam, ISBN 0-444-70259-8 (v. 1), 1987

[Coh89] L.D. Cohen and I. Cohen, *Finite-element methods for active contour models and balloons for 2-D and 3-D images*, IEEE Trans. Pattern Analysis and Machine Intelligence, 15(11):1131-1147, 1989

[Col93] A. Collignon, D. Vandermeulen, P. Suetens, G. Marchal, A. Baert, and A. Oosterlinck, *Surface based registration of 3d medical images*, Proc. SPIE Vol. 1898: Medical Imaging VII, 1993

[Col93b] A. Collignon, T. Geraud, D. Vandermeulen, P. Suetens, and G. Marchal, *New high-performance 3d registration algorithms for 3d medical images*, Proc. SPIE Vol. 1898: Image Processing, 779-788, 1993

[Col95] A. Collignon, D. Vandermeulen, P. Suetens, and G. Marchal, *3d multimodality image registration using features space clustering*, Proc. Computer Vision, Virtual Reality, and Robotics in Medicine (CVRMed'95), 195-204, 1995

[Col95b] A. Collignon, F. Maes, D. Delaere, D. Vandermeulen, P. Suetens, and G. Marchal, *Automated multi-modality image registration based on information theory*, Proc. Information Processing in Medical Imaging, 263-274, 1995

[Cli92] D.L. Collins, T.M. Peters, W. Dai, A.C. Evans, *Model-based segmentation of individual brain structures from MRI data*, Proc. SPIE Visualization in Biomedical Computing (1808), pp. 10-23, 1992

[Cnr80] R.W. Conners and C.A. Harlow, *A theoretical comparison of texture algorithms*, IEEE Trans. Pattern Analysis and Machine Intelligence, 2(3):204-222, 1980

[Cnr84] R.W. Conners, M.M. Trivedi, and C.A. Harlow, *Segmentation of a high-resolution urban scene using texture operators*, Computer Vision, Graphics, and Image Processing, 25:273-310, 1984

[Con84] K. Conradsen, *En introduktion til statistik* (Danish), IMM lecture notes, 1984

[Coo95] T.F. Cootes , D.H. Cooper, C.J. Taylor and J. Graham, *Active Shape Models - Their training and application*, Computer Vision Graphics and Image Processing, 61:38-59, 1995

[Cot96] S. Cotin, H. Delingette, M. Bro-Nielsen, N. Ayache, J.M. Clément, V. Tassetti and J. Marescaux, *Geometric and Physical representations for a simulator of hepatic surgery*, Proc. Medicine Meets Virtual Reality, 1996

[Cot96b] S. Cotin, H. Delingette, J.M. Clement, V. Tassetti, J. Marescaux, and N. Ayache, *Volumetric deformable models for simulation of laparascopic surgery*, Proc. Computer Assisted Radiology (CAR'96), pp. 793-798, 1996

[Cov93] S.A. Cover, N.F. Ezquerra and J.F. O'Brien, R. Rowe, T. Gadacz and E. Palm, *Interactively deformable models for surgery simulation*, IEEE Computer Graphics & Applications, pp. 68-75, Nov. 1993

[Dav96] C. Davatzikos, *Nonlinear registration of brain images using deformable models*, Proc. IEEE Workshop on Mathematical Methods in Biomedical Image Analysis, 1996

[Dav96b] C. Davatzikos, *Spatial normalization of 3d brain images using deformable models*, Submitted Journal of Computer Assisted Tomography, 1996

[Dec95] J. Declerck, G. Subsol, J-P. Thirion, and N. Ayache, *Automatic retrieval of anatomical structures in 3D medical images*, Proc. Computer Vision, Virtual Reality and Robotics in Medicine (CVRMed'95), pp; 153-162, 1995

[Dec96] J. Declerck, J. Feldmar, M.L. Goris, and F. Betting. *Automatic registration and alignment on a template of cardiac stress & rest SPECT images*, INRIA Tech. Rep. 2770, 1996

[Del94] H. Delingette, G. Subsol, S. Cotin and J. Pignon, *A craniofacial surgery simulation testbed*, INRIA Tech. Rep. 2199, 1994

[Del94b] H. Delingette, *Simplex Meshes: a general representation for 3D shape reconstruction*, Proc. Computer Vision and Pattern Recognition (CVPR'94), 1994

[Deu95] O. Deussen, L. Kobbelt, and P. Tücke, *Using simulated annealing to obtain good approximations of deformable bodies*, Proc. EuroGraphics Workshop on Animation and Simulation, 1995

[Doc76] M.P. Do Carmo, *Differential geometry of curves and surfaces*, USA, Prentice-Hall, 1976

[Du73] R.O. Duda and P.E. Hart, *Pattern classification and scene analysis*, Stanford Research Institute, Menlo Park, CA, USA, Wiley-Interscience, 1973

[Els91] P.A. van den Elsen, and M.A. Viergever, *Marker guided registration of electromagnetic dipole data with tomographic images*, Proc. Information Processing in Medical Imaging, pp. 142-153, 1991

[Els92] P.A. van den Elsen, J.B.A. Maintz, E-J.D. Pol, and M.A. Viergever, *Image fusion using geometrical features*, Proc. SPIE Vol. 1808: Visualization in Biomedical Computing, 172-186, 1992

[Els93] P.A. van den Elsen, E-J.D. Pol, and M.A. Viergever, *Medical image matching: A review with classification*, IEEE Eng. Med. Biol., 12:26-39, 1993

[Els94] P.A. van den Elsen, *Retrospective fusion of CT and MR brain images using mathematical operators*, AAAI Spring Symposium Series: Applications of Computer Vision in Medical Image Processing, 30-33, 1994

[Els94b] P.A. van den Elsen, T.S. Sumanaweera, P.F. Hemler, S. Napel, and J. Adler, *Grey value correlation techniques used for automatic matching of CT and MR brain and spine images*, Proc. SPIE Vol. 2359: Visualization in Biomedical Computing, 227-237, 1994

[Els95] P.A. van den Elsen, J.B.A. Maintz, E-J.D. Pol, and M.A. Viergever, *Automatic registration of CT and MR brain images using correlation of geometrical features*, IEEE Trans. on Medical Imaging, 14(2):384-396, 1995

[Eva88] A.C. Evans, C. Beil, S. Marret, C.J. Thompson, and A. Hakim, *Anatomical-functional correlation using an adjustable mri-based region of interest atlas with positron emission tomography*, J. Cerebral Blood Flow and Metabolism, 8:513-530, 1988

[Eva91] A.C. Evans, W. Dai, L. Collins, P. Neelin, S. Marret, *Warping of a computerized 3-D atlas to match brain image volumes for quantitative neuroanatomical and functional analysis*, Proc. SPIE Medical Imaging V (1445), pp. 236-246, 1991

[Eva92] A.C. Evans, S. Marret, P. Neelin, D.L. Collins, K. Worsley, W. Dai, S. Milot, E. Meyer, and D. Bub, *Anatomical mapping of functional activation in stereotactic coordinate space*, Neuroimage, 1:43-53, 1992

[Fab88] T.L. Faber and E.M. Stokely, *Orientation of 3-D structures in medical images*, IEEE Trans. Pattern Analysis and Machine Intelligence, 10:626-633, 1988

[Fau86] O.D. Faugeras and M. Hebert, *The representation, recognition, and location of 3-D objects*, Int. J. Robotics Research, 5:27-52, 1986

[Fel94] J. Feldmar and N. Ayache, *Rigid, affine, and locally affine registration of free-form surfaces*, Int. J. of Computer Vision, in press, also INRIA Tech. Rep. 2220, 1996

[Fid94] M. Fidrich and J-P. Thirion, *Multiscale extraction and representation of features from medical images*, INRIA Tech. Rep. 2365, 1994

[Fox85] P.T. Fox, J.S. Perlmutter, and M.E. Raichle, *A stereotactic method of anatomical localization for positron emission tomography*, J. Computer Assisted Tomography, 9(1):141-153, 1985

[GA86] A. Gamboa-Aldeco and G.T.Y. Chen, *Correlation of 3D surfaces from multiple modalities in medical imaging*, Proc. SPIE 626: Medicine XIV/PACS IV, pp. 467-473, 1986

[Ge92] Y. Ge, J.R. Votaw, R.A. Margolin, J.M. Fitzpatrick, R.J. Maciunas, and R.M. Kessler, *Reformatting PET images by fitting of the proportional grid system: Implementation anbd validation*, Proc. SPIE 1652: Medical Imaging VI: Image Processing, pp. 397-408, 1992

[Gee93] J.C. Gee, M. Reivich and R. Bajcsy, *Elastically deforming atlas to match anatomical brain images*, Journal of Computer Assisted Tomography, 17(2):225-236, March 1993

[Gei93] B. Geiger, *Three-dimensional modeling of human organs and its application to diagnosis and surgical planning*, INRIA Tech. Rep. 2105, Dec. 1993

[Gib95] S.F. Frisken Gibson, *Beyond volume rendering: Visualization, haptic exploration, and physical modeling of voxel-based objects*, Proc. EuroGraphics Workshop on Visualization in Scientific Computing, 1995

[Goo82] D.H. Goose and J. Appleton, *Human dentofacial growth*, Pergamon Press, ISBN 0-08-026393-3, 1982

[Gou89] J-P. Gourret, N.M. Thalmann and D. Thalmann, *Simulation of object and human skin deformations in a grasping task*, 23(3):21-30, 1989

[Gra96] C. Gramkow, *Registration of 2D and 3D medical images*, M.Sc. thesis, Department of Mathematical Modelling, Technical University of Denmark (IMM/DTU), 1996

[Gre91] Greitz, Bohm, Holte, and Eriksson, *A computerized brain atlas: Construction, anatomical content, and some applications*, J. Computer Assisted Tomography, 15(1):26-38, 1991

[Gue92] A. Gueziec and N. Ayache, *Smoothing and matching of 3-D space curves*, Proc. SPIE 1808: Visualization in Biomedical Imaging, pp. 259-273, 1992

[Har73] R.M. Haralick, K. Shanmugan, and I. Dinstein, *Textural features for image classification*, IEEE Trans. on Systems, Man, and Cybernetics, 3:610-621, 1973

[Har79] R.M. Haralick, *Statistical and structural approaches to texture*, Proc. of the IEEE, 67(5):786-804, 1979

[Haw90] D.J. Hawkes, D.L.G. Hill, E.D. Lehmann, G.P. Robinson, M.N. Maisey and A.C.F. Colchester, *Preliminary work on the interpretation of SPECT images with the aid of registered MR images and an MR derived 3D neuro-anatomical atlas*, in K.H. Höhne, *3D Imaging in Medicine*, NATO ASI series, Vol. F60, Springer-Verlag Berlin Heidelberg, 1990

[Her94] M. van Herk and H.M. Kooy, *Automatic three-dimensional correlation of CT-CT, CT-MRI, and CT-SPECT using chamfer matching*, Med. Phys. 21(7):1163-1178, 1994

[HT95] HT Medical, http://www.ht.com, 1995

[Hi91] D.L.G. Hill, D.J. Hawkes, J.E. Crossman, M.J. Gleeson, T.C.S. Cox, E.E.C.M.L. Bracey, A.J. Strong, and P. Graves, *Registration of MR and CT images for skull base surgery using point-like anatomical features*, The British Journal of Radiology, 64:1030-1035, 1991

[Hi92] D.L.G. Hill, S.E.M. Green, J.E. Crossman, D.J. Hawkes, G.P. Robinson, C.F. Ruff, T.C.S. Cox, A.J. Strong, and M.J. Gleeson, *Visualisation of multi-modal images for planning of skull base surgery*, Proc. SPIE 1808: Visualization in Biomedical Computing, pp. 564-573, 1992

[Hi92b] D.L.G. Hill, D.J. Hawkes, Z. Hussain, S.E.M. Green, C.F. Ruff, and G.P. Robinson, *Accurate combination of CT and MR data of the head: Validation and applications in surgical and threapy planning*, in C. Roux, G.T. Herman, and R. Collorec, *3D advanced image processing in medicine*, pp. 79-83, 1992

[Hi93] D.L.G. Hill, *Combination of 3D medical images from multiple modalities*, Ph.D. thesis, University of London, 1993

[Hi93b] D.L.G. Hill, D.J. Hawkes, N. Harrison, and C.F. Ruff, *A strategy for automated multimodality registration incorporating anatomical knowledge and imager characteristics*, Proc. Information Processing in Medical Imaging, Lecture Notes in Computer Science 687, Springer-Verlag, Berlin, 182-196, 1993

[Hi94] D.L.G. Hill and D.J. Hawkes, *Medical image registration using knowledge of adjacency of anatomical structures*, Image and Vision Computing, 12:173-178, 1994

[Hi94b] D.L.G. Hill and D.J. Hawkes, *Medical image registration using voxel similarity measures*, AAAI Spring Symposium Series: Applications of Computer Vision in Medical Image Processing, 34-37, 1994

[Hi94c] D.L.G. Hill, C. Studholme, and D.J. Hawkes, *Voxel similarity measures for automated image registration*, Proc. Visualization in Biomedical Computing, 205-216, 1994

[Hoe92] K.H. Höhne, M. Bomans, M. Riemer, R. Schubert, U. Tiede and W. Lierse, *A 3D anatomical atlas based on a volume model*, IEEE Computer Graphics Applications, 12(4):72-78, 1992

[Hoe92b] K.H. Höhne, A. Pommert, M. Riemer, T. Schiemann, R. Schubert, and U. Tiede, *Framework for the generation of 3d anatomical atlases*, Proc. SPIE 1808: Visualization in Biomedical Computing, pp. 510-520, 1992

[Hoe94] K.H. Höhne, A. Pommert, M. Riemer, T. Schiemann, R. Schubert, and U. Tiede, *Representing spatial knowledge using the intelligent volumes concept*, Proc. Am. Ass. Art. Int., Spring Symposium: Applications of Computer Vision in Medical Image Processing, 1994

[Hor87] B.K.P. Horn, *Closed-form solution of absolute orientation using unit quaternions*, J. Opt. Soc. Amer. A, 4:629-642, 1987

[Hor88] B.K.P. Horn, H.M. Hilden, and S. Negahdaripour, *Closed-form solution of absolute orientation using orthonormal matrices*, J. Opt. Soc. Amer. A, 5:1127-1135, 1988

[Hue75] K.H. Huebner, *The finite element method for engineers*, John Wiley & Sons, ISBN 0-471-41950-8, 1975

[Jia92] H. Jiang, K.S. Holton, and R.A. Robb, *Image registration of multimodality 3-D medical images by chamfer matching*, Proc. SPIE Vol. 1660: Biomedical Image Processing and Three-dimensional Microscopy, 356-366, 1992

[Jia92b] H. Jiang, R.A. Robb, and K.S. Holton, *A new approach to 3-d registration of multimodality medical images by surface matching*, Proc. SPIE Vol. 1808: Visualization in Biomedical Computing, 196-213, 1992

[Kar87] H. Kardestuncer, *Finite element handbook*, McGraw-Hill, ISBN 0-07-033305-X, 1987

[Kas88] M. Kass, A. Witkin and D. Terzopoulos: *Snakes: Active contour models*, Int. J. of Computer Vision, 2:321-331, 1988

[Kee95] E. Keeve, S. Girod, P. Pfeifle, and B. Girod, *Interactive craniofacial surgery planning by 3D-simulation and visualization*, Proc. Int. Workshop on Rapid Prototyping in Medicine & Computer-Assisted Surgery, 1995

[Kee95b] E. Keeve, S. Girod, J. Jauch, and B. Girod, *Anatomy-based modeling of human facial tissue for craniofacial surgery simulation*, Proc. Visualization'95 - Dynamics and Complexity, 1995

[Kee96] E. Keeve, S. Girod, P. Pfeifle, and B. Girod, *Anatomy-based facial tissue modeling using the finite element method*, to appear Visualization'96, 1996

[Kee96b] E. Keeve, S. Girod, and B. Girod, *Craniofacial surgery simulation*, to appear Visualization in Biomedical Computing (VBC'96), 1996

[Kee96c] E. Keeve, S. Girod, and B. Girod, *Computer-aided craniofacial surgery*, Proc. Computer Assisted Radiology (CAR'96), pp. 757-762, 1996

[Ker96] J. Kerr, P. Ratiu, and M. Sellberg, *Volume Rendering of Visible Human Data for an Anatomical Virtual Environment*, Proc. Medicine Meets Virtual Reality IV, or http://www.eai.com/interactive/dhuman/paper/paper.html, 1996

[Koc96] R.M. Koch, M.H. Gross, F.R. Carls, D.F. von Büren, G. Fankhauser, and Y.I.H. Parish, *Simulating facial surgery using finite element models*, to appear SIGGRAPH, 1996

[Ku96] C. Kuhn, U. Kühnapfel, H.-G. Krumm, B. Neisius, *A 'virtual reality' based training system for minimally invasive surgery*, Proc. Computer Assisted Radiology (CAR'96), pp. 764-769, 1996

[Kue93] U.G. Kühnapfel, B. Neisius, H.G. Krumm, and M. Hübner, *CAD-based simulation and modelling for endoscopic surgery*, Endoscopic Surgery, 1993

[Kue94] U.G. Kühnapfel and B. Neisius, *Realtime graphical computer simulation for endoscopic surgery*, Proc. Medicine Meets Virtual Reality II, 1994

[Kue95] U.G. Kühnapfel, H. Krumm, C. Kuhn, M. Hübner, and B. Neisius, *Endosurgery simulation with kismet*, Proc. Virtual Reality World'95, 1995

[Lee93] Y. Lee, D. Terzopoulos, and K. Waters, *Constructing physical-based facial models of individuals*, Proc. Graphics Interface'93, pp. 1-8, 1993

[Lee95] Y. Lee, D. Terzopoulos, and K. Waters, *Realistic modeling of facial animation*, Proc. SIGGRAPH'95, 1995

[Lem91] D. Lemoine, C. Barillot, B. Gibaud, and E. Pasqualini, *An anatomical-based 3d registration system of multimodality and atlas data in neurosurgery*, Proc. Information Processing in Medical Imaging, pp. 154-164, 1991

[Lev88] D.N. Levin, C.A. Pelizarri, G.T.Y. Chen, C-T. Chen and M.D. Cooper, *Retrospective geometric correlation of MR, CT, and PET images*, Radiology, 169:817-823, 1988

[Lin86] F.P.G.M. Linden, *Facial growth and facial orthopedics*, Quintessence Publishing Co., Chicago, 1986

[Lo94] L.J. Lo, J.L. Marsh, M.W. Vannier, and V.V. Patel, *Craniofacial computer assisted surgical planning and simulation*, Clin. Plast. Surg., 21:501-516, 1994

[Lor87] W.E. Lorensen and H. Cline, *Marching Cubes: A high resolution 3D surface construction algorithm*, Computer Graphics, 21(4):163-169, July 1987

[Mau93] C.R. Maurer and J.M. Fitzpatrick, *A review of medical image registration*, in R.J. Maciunas, *Interactive image-guided neurosurgery*, Park Ridge, IL: American Association of Neurological Surgeons, 17-44, 1993

[McV85] E.R. McVeigh, R.M. Henkelman and M.J. Bronskill, *Noise and filtration in magnetic resonance imaging*, Medical Physics, 12(5):586-591, 1985

[Mil93] M.I. Miller, G.E. Christensen, Y. Amit and U. Grenander, *Mathematical textbok of deformable neuroanatomies*, Proc. Natl. Acad. Sci. USA, 90:11944-11948, December 1993

[Mon92] O. Monga, S. Benayoun, and O.D. Faugeras, *From partial derivatives of 3D density images to ridge lines*, Proc. SPIE 1808: Visualization in Biomedical Imaging, pp. 118-129, 1992

[Mos94] M. Moshfeghi, S. Ranganath, and K. Nawyn, *Three-dimensional elastic matching of volumes*, IEEE Trans. Image Processing, 3(2):128-138, 1994

[Nie94] M. Nielsen, L. Florack and R. Deriche, *Regularization and Scale Space*, INRIA Tech. Rep. RR-2352, September 1994

[Nie96] M. Nielsen, R. Maas, W. Niessen, L. Florack, and B. ter Haar Romeny, *Binocular stereo from grey-scale images*, Submitted for Journal of Mathematical Imaging and Vision, 1996

[Nie96b] M. Nielsen, *Personal communication*, 1996

[Pel89] C.A. Pelizarri, G.T.Y. Chen, D.R. Spelbring, R.R. Weichselbaum and C-T. Chen, *Accurate three-dimensional registration of CT, PET, and/or MR images of the brain*, Journal of Computer Asssisted Tomography, 13(1):20-26, 1989

[Pen96] X. Pennec and N. Ayache, *Randomness and geometric features in computer vision*, INRIA Tech. Rep. 2820, short version submitted to CVPR'96, 1996

[Pet86] T.M. Peters, J.A. Clark, A. Olivier, E.P. Marchand, G. Mawko, M. Dieumegarde, L.V. Muresan and R. Ethier, *Integrated stereotaxic imaging with CT, MR imaging, and digital subtraction angiography*, Radiology, 161:821-826, 1986

[Pie92] S.D. Pieper, *CAPS: Computer-aided plastic surgery*, Ph.D. thesis, MIT, 1992

[Pie92b] S.D. Pieper, J. Rosen, and D. Zeltzer, *Interactive graphics for plastic surgery*, Computer Graphics, 26(2):127-134, 1992

[Pok96] P. Pokrandt, *Fast non-supervised matching: A probabilistic approach*, Proc. Computer Assisted Radiology (CAR'96), pp. 306-310, 1996

[Pow64] M.J.D. Powell, *An efficient method for finding the minimum of a function of several variables without calculating derivatives*, Comput. J., 7:155-163, 1964

[Num92] W.H. Press, S.A. Teukolsky, W.T. Vetterling, and B.P. Flannery, *Numerical recipes in C*, Cambridge University Press, 1992

[Rab95] R.D. Rabbitt, J.A. Weiss, G.E. Christensen and M.I. Miller, *Mapping of hyperelastic deformable templates using the finite element method*, Proc. SPIE vol. 2573, pp.252-265, 1995

[Ras96] B. Rasmussen, *Cranio-facial surgery simulation*, IMM M.Sc. Thesis, 1996

[Sag94] M.A. Sagar, D. Bullivant, G.D. Mallinson, P.J. Hunter and I. Hunter, *A virtual environment and model of the eye for surgical simulation*, Proc. SIGGRAPH'94, pp. 205-212, 1994

[Sa93] R.M. Satava, *Virtual reality surgical simulator*, Surgical Endoscopy, 7:203-205, 1993

[Sat92] J. Satoh, H. Ciyokura, M. Kobayashi and T. Fujino, *Simulation of surgical operations based on solid modelling*, in T. L. Kunii editor, *Visual computing, Integrating computer graphics with computer vision*, Tokyo (Japan), Computer Graphics Society, Springer Verlag, pp. 907-916, 1992

[Smi96] B. Smith, P. Bjorstad, and W. Gropp, *Domain Decomposition: Parallel Multilevel Methods for Elliptic PDES*, Cambridge University Press, ISBN 0-521-49589-X, 1996

[Ste92] D.E. Stewart, *Meschach: Matrix computations in C*, WWW ftp://ftpmaths.anu.edu.au/pub/meschach/meschach.html, 1992

[Stu95] C. Studholme, D.L.G. Hill and D.J. Hawkes, *Multiresolution voxel similarity measures for MR-PET registration*, Proc. Information Processing in Medical Imaging, 287-298, 1995

[Stu96] C. Studholme, D.L.G. Hill and D.J. Hawkes, *Automated 3D registration of truncated MR and CT images of the head*, Proc. British Machine Vision Conference (BMVC'96), pp. 27-37, 1996

[Stu96b] C. Studholme, D.L.G. Hill and D.J. Hawkes, *Automated 3D registration of MR and CT images of the head*, Medical Image Analysis, 1(2), 1996

[Sub95] G. Subsol, J-P. Thirion, and N. Ayache, *A general scheme for automatically building 3D morphometric anatomical atlases: Applications to a skull atlas*, Proc. Medical Robotics and Computer Assisted Surgery (MRCAS'95), pp. 226-233, 1995

[Sub96] G. Subsol, J-P. Thirion, and N. Ayache, *Some medical applications of an automatically built 3D morphometric skull atlas*, Proc. Computer Assisted Radiology (CAR'96), pp. 339-344, 1996

[Sze94] R. Szeliski and S. Lavallee, *Matching 3-D anatomical surfaces with non-rigid deformations using octree-splines*, Proc. CVPR Workshop on Biomedical Image Analysis, pp. 144-153, June 1994

[Tal88] J. Talairach and P. Tournoux, *Co-planar stereotactic atlas of the human brain* George Thieme Verlag, Stuttgart, 1988

[Tan95] I. Tanaka, M. Kobayashi, T. Fujino, and H. Chiyokura, *Simulation for facial lip expression using the facial muscle model*, Proc. Computer Assisted Radiology (CAR'95), pp. 878-881, 1995

[Tay92] R.H. Taylor, *An overview of computer assisted surgery research at IBM T.J. Watson research center*, in *Innovation et technologie en biologie et medecine*, 1992

[Ter88] D. Terzopoulos and K. Fleischer, *Deformable models*, The Visual Computer, 4:306-331, 1988

[Ter91] D. Terzopoulos and D. Metaxas, *Dynamic 3D models with local and global deformations: Deformable superqudraics*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 13(7):703-714, 1991

[Ter91b] D. Terzopoulos and K. Waters, *Techniques for realistic facial modeling and animation*, Computer Animation, Springer, pp. 59-73, 1991

[Ter93] D. Terzopoulos and K. Waters, *Analysis and synthesis of facial image sequences using physical and anatomical models*, IEEE Trans. on Pattern Analysis and Machine Intelligence, 15(6):569-579, 1993

[Thi92] J-P. Thirion and A. Gourdon, *The 3d marching lines algorithm and its application to crest lines extraction*, INRIA Tech. Rep. 1672, 1992

[Thi92b] J-P. Thirion, A. Gourdon, O. Monga, A. Gueziec, and N. Ayache, *Fully automatic registration of 3D CAT-scan images using crest lines*, Proc. Annu. Int. Conf. IEEE Eng. Med. Biol. Soc., 14:1888-1890, 1992

[Thi93] J-P. Thirion and A. Gourdon, *The 3d marching lines algorithm: New results and proofs*, INRIA Tech. Rep. 1881, 1993

[Thi93b] J-P. Thirion, *The extremal mesh and the understanding of 3D surfaces*, INRIA Tech. Rep. 2149, 1993

[Thi96] J-P. Thirion, *Non-rigid matching using demons*, Proc. Int. Conf. Computer Vision and Pattern Recognition (CVPR'96), 1996

[Tie96] U. Tiede, T. Schiemann, and K. H. Hoehne, *Visualizing the Visible Human*, IEEE Comput. Graphics Appl., 16(1):7-9, 1996

[Tsu89] Y. Tsumiyama, K. Sakaue and K. Yamamoto, *Active net : Active net model for region extraction* (in Japanese), IPSJ SIG Notes, CV 63-2, 89(96):1-8, 1989

[Und92] P.E. Undrill, G.G. Cameron, M.J. Cookson, C. Davies, N.L. Robinson, A. Hill, T.F. Cootes, C.J. Taylor, A. Thornhan, J. Wysocki, H.M. Liddell, and D. Parkinson, *Integrated presentation of 3d data derived from multi-sensor imagery and anatomical atlases using a parallel processing system*, Proc. SPIE 1653: Image Capture, Formatting and Display, 1992

[Vio95] P. Viola and W.M. Wells III, *Alignment by maximization of mutual information*, Proc. Int. Conf. Computer Vision, 1995

[Vis96] *The Visible Human Project*, WWW http://www.nlm.nih.gov/research/visible/visible_human.html, 1996

[Wat87] K. Waters, *A muscle model for animating three-dimensional facial expression*, Computer Graphics, 21(4):17-24, 1987

[Wat91] K. Waters and D. Terzopoulos, *Modeling and animating faces using scanned data*, J. Visualization Comput. Animation, 2(4):123-128, 1991

[Wat92] K. Waters, *A physical model of facial tissue and muscle articulation derived from computer tomography data*, Proc. Visualization in Biomedical Computing (VBC'96), pp. 574-583, 1992

[Wel96] W.M. Wells III, P. Viola, H. Atsumi, S. Nakajima, and R. Kikinis, *Multi-modal volume registration by maximization of mutual information*, Medical Image Analysis, 1996

[Wes96] J. West, J.M. Fitzpatrick, et al., *Comparison and evaluation of retrospective intermodality image registration techniques*, Proc. SPIE Vol. 2710: Medical Imaging - Image Processing, 1996

[Wlh96] J. Wilhelms, *Modeling animals with bones, muscles, and skin*, to appear IEEE Computer Graphics and Applications, 1996

[Wil92] D.J. Williams and M. Shah, *A fast algorithm for active contours and curvature estimation*, CVGIP: Image Understanding, 55(1):14-26, 1992

[Woo92] R.P. Woods, S.R. Cherry, and J.C. Mazziotta, *Rapid automated algorithm for aligning and reslicing PET images*, Journal of Computer Assisted Tomography, 16(4):620-633, 1992

[Woo93] R.P. Woods, J.C. Mazziotta, and S.R. Cherry, *MRI-PET registration with automated algorithm*, Journal of Computer Assisted Tomography, 17(4):536-546, 1993

[Wyv86] G. Wywill, C. McPheeters and B. Wyvill, *Data structure for soft objects*, The Visual Computer, 2:227-234, 1986

[Yas90] T. Yasuda, Y. Hashimoto, S. Yokoi and J-I. Toriwaki, *Computer system for craniofacial surgical planning based on CT images*, IEEE Trans. Medical Imaging, 9(3):270-280, 1990

[Zha90] J. Zhang, M.F. Levesque, C.L. Wilson, R.M. Harper, J. Engel, Jr., R. Lufkin, and E.J. Behnke, *Multimodality imaging of brain structures for stereotactic surgery*, Radiology, 175:435-441, 1990

[Zha94] Z. Zhang, *Iterative point matching for registration of free-form curves and surfaces*, Int. J. Computer Vision, 13(2):119-152, 1994

Ph. D. theses from IMM

1. **Larsen, Rasmus.** (1994). *Estimation of visual motion in image sequences. xiv* + 143 pp.

2. **Rygard, Jens Moberg.** (1994). *Design and optimization of flexible manufacturing systems. xiii* + 232 pp.

3. **Lassen, Niels Christian Krieger.** (1994). *Automated determination of crystal orientations from electron backscattering patterns. xv* + 136 pp.

4. **Melgaard, Henrik.** (1994). *Identification of physical models. xvii* + 246 pp.

5. **Wang, Chunyan.** (1994). *Stochastic differential equations and a biological system. xxii* + 153 pp.

6. **Nielsen, Allan Aasbjerg.** (1994). *Analysis of regularly and irregularly sampled spatial, multivariate, and multi-temporal data. xxiv* + 213 pp.

7. **Ersbøll, Annette Kjær.** (1994). *On the spatial and temporal correlations in experimentation with agricultural applications. xviii* + 345 pp.

8. **Møller, Dorte.** (1994). *Methods for analysis and design of heterogeneous telecommunication networks.* Volume 1-2, *xxxviii* + 282 pp., 283-569 pp.

9. **Jensen, Jens Christian.** (1995). *Teoretiske og eksperimentelle dynamiske undersøgelser af jernbanekøretøjer.* ATV Erhvervsforskerprojekt EF 435. *viii* + 174 pp.

10. **Kuhlmann, Lionel.** (1995). *On automatic visual inspection of reflective surfaces.* ATV Erhvervsforskerprojekt EF 385. Volume 1, *xviii* + 220 pp., (Volume 2, *vi* + 54 pp., fortrolig).

11. **Lazarides, Nikolaos.** (1995). *Nonlinearity in superconductivity and Josephson Junctions. iv* + 154 pp.

12. **Rostgaard, Morten.** (1995). *Modelling, estimation and control of fast sampled dynamical systems. xiv* + 348 pp.

13. **Schultz, Nette.** (1995). *Segmentation and classification of biological objects. xiv* + 194 pp.

14. **Jørgensen, Michael Finn.** (1995). *Nonlinear Hamiltonian systems. xiv* + 120 pp.

15. **Balle, Susanne M.** (1995). *Distributed-memory matrix computations. iii* + 101 pp.

16. **Kohl, Niklas.** (1995). *Exact methods for time constrained routing and related scheduling problems. xviii* + 234 pp.

17. **Rogon, Thomas.** (1995). *Porous media: Analysis, reconstruction and percolation. xiv* + 165 pp.

18. **Andersen, Allan Theodor.** (1995). *Modelling of packet traffic with matrix analytic methods. xvi* + 242 pp.

19. **Hesthaven, Jan.** (1995). *Numerical studies of unsteady coherent structures and transport in two-dimensional flows.* Risø-R-835(EN) 203 pp.

20. **Slivsgaard, Eva Charlotte.** (l995). *On the interaction between wheels and rails in railway dynamics. viii* + 196 pp.

21. **Hartelius, Karsten.** (1996). *Analysis of irregularly distributed points. xvi* + 260 pp.

22. **Hansen, Anca Daniela.** (1996). *Predictive control and identification - Applications to steering dynamics. xviii* + 307 pp.

23. **Sadegh, Payman.** (1996). *Experiment design and optimization in complex systems. xiv* + 162 pp.

24. **Skands, Ulrik.** (1996). *Quantitative methods for the analysis of electron microscope images. xvi* + 198 pp.

25. **Bro-Nielsen, Morten.** (1996). *Medical image registration and surgery simulation. xxvii* + 274 pp.